AN ADAPTIVE TIMEOUT SCHEME FOR SHORT TCP FLOWS OVER 3G WIRELESS NETWORKS

T V Prabhakar, Joy Kuri

CEDT, Indian Institute of Science,
Bangalore - 560012,
INDIA
tvprabs, kuri@cedt.iisc.ernet.in

Kiran K N[+]*, Kartik M*[*]

[+]Dept. of CS, IIT Delhi,
[*]Dept. of CS, UCL
kiran@cse.iitd.ernet.in
K.Muralidharan@cs.ucl.ac.uk

ABSTRACT

The focus of this work is on performance enhancement of short-lived TCP flows over a wide-area network with the last hop being a 3G wireless link. We take the 3GPP Non Real Time traffic model and show that when such a short-lived flow encounters a prolonged bad period on the wireless link, the TCP source's congestion window can become full, causing the source to stall, waiting for ACKs that never arrive. The stalled source then wastes time till the RTO timer expires, causing a timeout. The effect of this is poor TCP throughput. Our work tackles this specific problem by using two mechanisms: (a) a feedback mechanism from the base station back to the TCP source using ICMP messages indicating severe bad periods on the wireless link, and (b) an adaptive mechanism that adjusts the RTO timer value to avoid situations where the source idles unnecessarily. The RTO timer adjustment can lead to either *premature* or *delayed* timeouts, depending upon the state of the TCP source when the feedback arrives. Simulation results indicate that the proposed timeout mechanism offers over 100% throughput performance improvement even when the bad state is about 50%. The average RTO value is lower with our scheme, the average goodput obtained is 85% even at 50% bad state, and the source idle time is at least 3 times less with our scheme. The scheme performs well also for bulk data traffic.

I. RELATED WORK

In [1], experiments show that a reliable link layer protocol with some knowledge of TCP performs very well. The experiments indicate that shielding the TCP sender from duplicate acknowledgements caused by wireless losses improves throughput by 10-30%. This is achieved by exploiting the coarse grained and fine-grained timers of TCP and link layer protocols respectively. Also, the link layer is assumed to perform in order delivery across the link. A detailed study and evaluation is carried out for several end-to-end, split connection and link layer protocols. The performance metrics used are both throughput and goodput.

In this work, link layer solutions are of interest since they hide wireless deficiencies from TCP. The main problem with link layer recovery is the possibility of timer interactions and competing retransmissions between link layer ARQ and source TCP. In [2], simulation results show that link layer recovery causes TCP sources to effectively increase the Round Trip Time (RTT) estimates and Retransmission Timeout (RTO) values. This increase in source timeout now decreases the likelihood of retransmissions. However, the paper considers only bulk data transfers. In [3], an Explicit Bad State Notification (EBSN) message is sent from base stations to TCP Tahoe sources in the event of bad wireless state. The source in turn resets its RTO value and ensures no timeout as long as the base station issues EBSNs. A high goodput is made possible due to lesser number of packet retransmissions. However, in the model, only bulk data transfers are considered. In [4] ICMP messages are conveyed to hosts about satellite link outage and subsequent link restoration. In [5] and [6], an attempt is made to differentiate between packet loss due to network congestion and a packet corruption on wireless link. While in [5], ICMP messages are used to carry this information to sources, in [6] explicit loss notification is issued by receivers. However, the traffic model again is for bulk data trans-

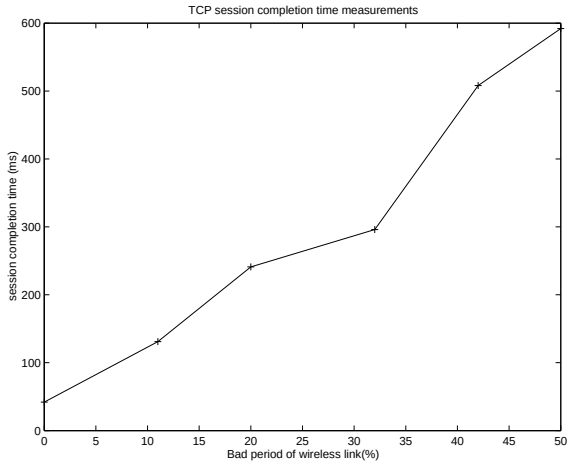

Fig. 1. *Session completion time for a short and bursty traffic flow.*

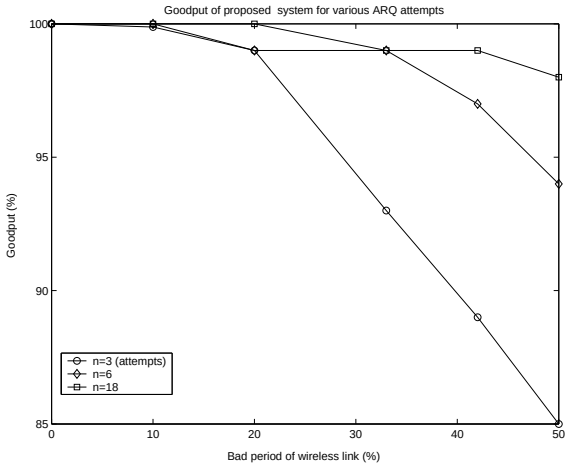

Fig. 2. *Goodput of TCP with 3 packet based ARQ link buffer.*

fer. In [7], it is mentioned that end-to-end performance is enhanced by the presence of ARQ since it takes over the error detection and correction. This permits TCP to ultimately act transparently on an error-free medium. The traffic model assumed is bulk data transfer.

I-A. Problem Statement

In this paper, our focus is on short-lived TCP flows. The problem statement is best captured by Figure 1. We define "session completion time" as the mean interval of time between the initiation and termination of data transfer on a single TCP session. We have performed simulations to show the effect of bad periods on the wireless link on the TCP session completion time for a single short and bursty traffic model. The objective is to obtain a reduced session completion time even when the wireless link sees severe bad periods. In [8], [9] it is reported that current analytical models assume arbitrarily long TCP connections where the complex and random behaviour of TCP can be studied. However, available evidence in [10] suggests that 95% of total traffic volume is TCP and over 80% are TCP flows. Among these, a majority are short connections – usually transferring under 10KB. Therefore, TCP rarely goes beyond slow start since the short transfer is quickly completed. However, RTT estimation itself can be unreliable due to its dependence on link quality. Also due to fewer RTT observations available for short flows, RTT estimates may not be reliable. Therefore, exponential increase of RTO is the actual source of long idle times during data transmissions on links with frequent errors or disconnections. Furthermore, the probability of fast retransmit algorithms getting invoked is also considerably low since at least three out-of-order packets are required to generate duplicate ACKs.

The problem of competing retransmission of packets both by a TCP source and link layer at the base station is of high significance. The link layer by its repeated attempts spends a sufficient amount of time, leading to a source timeout state. Goodput is a performance measure to evaluate the number of retransmissions. In [11], goodput is defined as the ratio of total number of bytes successfully received to the total number of bytes transmitted.

In general, a TCP segment can be fragmented into several link level frames and link level ARQ is concerned with retransmitting these frames for reliable transmissions. However, in our simulations we consider a large MTU for simplicity so that transmitted TCP segment fits into one link frame. From simulations, we obtain the best ARQ window and best number of ARQ attempts. We ensure that there are no competing retransmissions between the TCP source and the base station packet based link layer by flushing the link-level packet buffer after the specific number of retransmission attempts. Also, in-order delivery is implemented across the link.

Figure 2 shows the goodput for TCP flows with link level ARQ used in our simulations. With three at-

tempts, our performance metric was found to be about 85% even for 50% bad period. Having completed this initial link characterization, where we do not in anyway modify TCP, we proceed to propose an *extremely simple* mechanism for TCP. This modification applies a check on the RTO growth. In our scheme, a single message handles 5 cases and an optional message handles an additional 3 cases.

II. PROPOSED SOLUTION

We consider a bad period on the wireless link. If the link layer can report ARQ failure, then TCP sources can take their own appropriate decision on how to sustain a good throughput level. Such a decision may include *either to extend its* RTO with the hope that wireless link may improve or *even prematurely timeout* and invoke congestion related algorithms. However, this should be done without affecting the back-off algorithms.

The proposed solution relies on two mechanisms: (a) a feedback mechanism from the base station back to the TCP source using ICMP messages indicating severe bad periods on the wireless link, and (b) an adaptive mechanism that adjusts the RTO timer value to avoid situations where the source idles unnecessarily. The adaptive TCP source timeout mechanism is based on ICMP messages (ICMP_DROP and ICMP_DEFER) generated at the Radio Network Controllers (RNC). The mechanism may best be understood by way of such messages arriving at sources. Figure 3 captures the state transitions of a TCP source that are induced by our mechanism, including the cases automatically handled by the source.

ICMP_DROP: This message generated by a Base Station (Node B), informs the Radio Network Controller (RNC) of failure at the end of a fixed number of retries. Furthermore, this message conveys to the TCP source that the link layer ARQ window and buffer are completely flushed. The message indicates a long wireless bad period.

The response to the received ICMP_DROP message depends on the state of the TCP source. Let the current clock value at the source be denoted by b, and the RTO timer value by a. We consider possible cases (also numbered in Figure 3) below.

Case 1 - Immediate Timeout: If the difference $(a - b)$ is a small number, say, less than 25% of the RTO, i.e., $b > 3(a/4)$, the source times out at once. This gain in time helps in decreasing the idle time at the source.

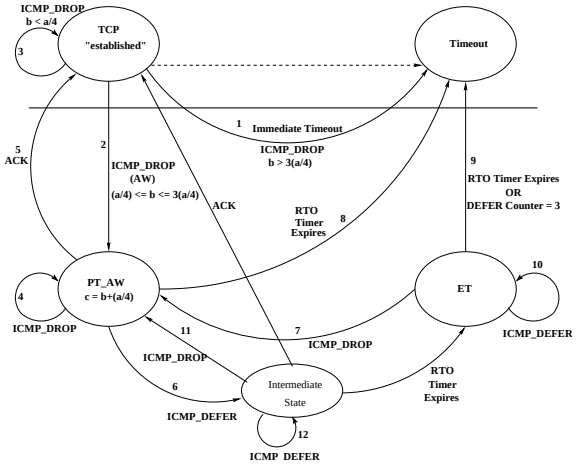

Fig. 3. *State diagram of TCP New Reno sources for the proposed scheme.*

Case 2 - Shrink Timeout: If $(a/4) \leq b \leq 3(a/4)$, the source recomputes a new timeout (c), by adding RTO/4 to its current clock value, i.e., $c = b + (a/4)$. Thus, the source is now set to timeout at c. The source enters "Premature Timeout" (PT) state.

Case 3 - Packets in flight: When b $< (a/4)$, the packets transmitted last from the source are probably still in flight. So, we ignore those ICMP_DROP messages which arrive when $b < (a/4)$. Therefore, our mechanism triggers only if the source clock has advanced to at least 25% of the RTO value.

Case 4 - Ignore drop: When the source is in state PT, we ignore new ICMP_DROP messages.

Case 5 - Restore timeout: Suppose an ACK arrives when the source is in state PT. Then the source exits PT and returns to normal new Reno state.

ICMP_DEFER: This message is generated by Node B and informs RNC either of the presence of at least 3 packets in the link buffer or ARQ success at the end of fixed number of selective repeat ARQ retries. The message indicates that the state of the wireless channel is improving.

Case 6 - Conditional extended timeout: If an ICMP_DEFER message arrives when the source is in state PT, we cancel the timer value and the state. The source moves to an Intermediate State as a step towards the "Extended Timeout"(ET) state. ET is entered *only* after RTO (a) has expired.

Case 7 - Exit extended timeout: If an ICMP_DROP ar-

rives when the source is in state ET, the source transits back to PT state and its associated timer value.

Case 8 and 9 Timeout - These cases occur upon TCP timer expiry.

Case 10 - Increment Defer Counter: If an ICMP DEFER arrives when source is in ET, the "DEFER Counter" is incremented by one. This counter can take a maximum value of 3.

Case 11 - Exit IS: If an ICMP_DROP message arrives when source is in Intermediate State, the source exits and returns to PT state.

Case 12 - Ignore Defer: If an ICMP DEFER message arrives when source is in Intermediate State, the source ignores such messages.

III. TRAFFIC AND WIRELESS LINK ERROR MODEL

The traffic model complies with the standard specified in [12]. For generating burst errors, we model the radio link by a two state Markov model. This model operates at the packet level. The average packet size as per the model is 480 Bytes. The good state BER is 10^{-6}, corresponding to a packet error rate of about 0.37% and bad state BER is 10^{-3}, corresponding to a packet error rate of 97.85%. The mean good period on the wireless link is set to 8 seconds and we vary the mean bad period from 0 to 8 seconds.

IV. SIMULATION SETUP

Simulations were carried out using Network Simulator (ns) version 1.1. We have modified the existing code used in [3] and used the experimental New Reno TCP. The timer granularity of 100 milliseconds was chosen. We simulate a single user running a single session. The setup consists of a wired sender transmitting packets to a forwarding base station. The base station forwards packets to the UMTS User Equipment (UE). In [13], the QoS profile for web page downloading is mentioned as having 64Kbps as the maximum bit rate. We have set the link bandwidth on both wired and wireless parts to 64Kbps with an RTT of 1 second.

V. RESULTS AND DISCUSSIONS

During a bad period, there is a window of failed packets waiting to complete the fixed number of ARQ attempts. After exhausting these attempts, we flush the ARQ buffer at the base station and source is informed using the ICMP_DROP message. We determine experimentally both the required packet level ARQ window size and the packet level ARQ attempts. Results indicate that it is sufficient to have a 3 packet ARQ buffer with at least 6 ARQ retransmission attempts. Using these results, we obtain the throughputs of TCP with and without our mechanism. We conduct measurements both for bulk and interactive traffic models. We also determine the average RTO value at the source with and without our scheme. Finally, we show the session completion time using our mechanism.

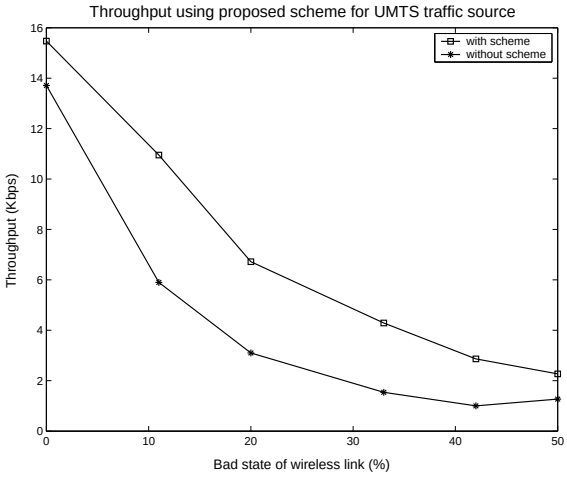

Fig. 4. *Throughput of TCP with and without the proposed scheme.*

V-A. Throughput measurements (UMTS source)

We now show the benefit our mechanism can provide with respect to the throughput performance. Figure 4 shows the results of the throughput performance for the traffic model described in the earlier section.

The effectiveness of our scheme for short TCP flows can be seen: even when the link is in the bad state 50% of the time, the throughput is about double that in default TCP.

V-B. RTO measurements for UMTS traffic source

We investigated the impact of our scheme on the RTO value (Figure 5). With our scheme, for a bad period of 11%, we obtain an average RTO of 9.90 seconds compared to 15 without our scheme. The RTO rises gradually as the fraction of bad period increases.

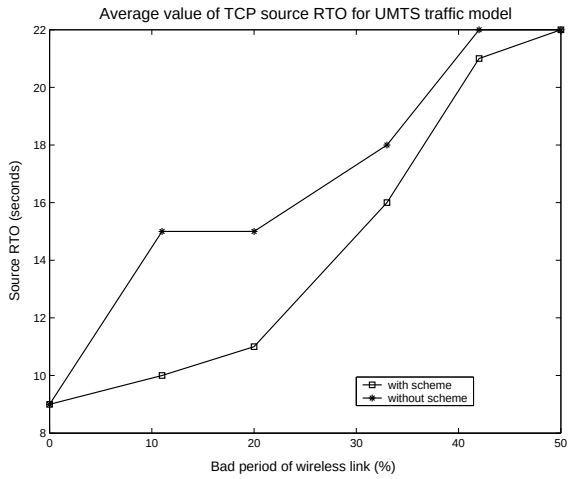

Fig. 5. *Average RTO (in seconds) for UMTS traffic source.*

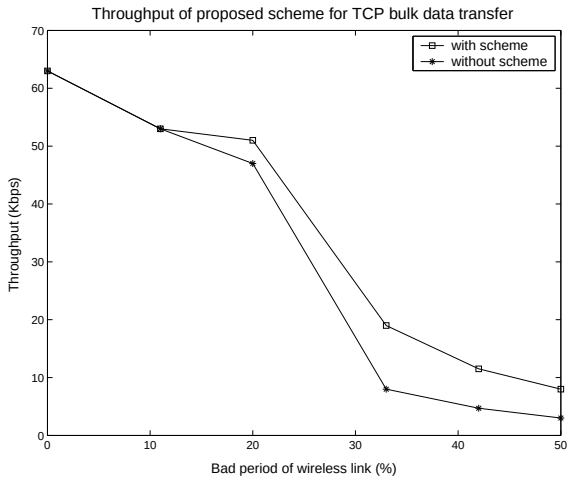

Fig. 6. *Throughput of TCP for a bulk data traffic source.*

V-C. Throughput measurements (Bulk Transfer)

Figure 6 shows the throughput performance of an FTP-like source for fixed packet size of 1500 bytes. Significant benefits are observed for a bad period of 33%.

V-D. RTO measurements for ftp traffic source

Figure 7 shows the RTO build-up for an FTP data source. Significant benefits of a lower RTO may be seen only beyond 33% bad period.

V-E. Session completion time comparisons

Figure 8 shows the session completion time with and without the proposed scheme. We compare the session completion time when the channel is good throughout, with that when the channel is in a bad state. It can be seen from Figure 8 that for any given "fraction of bad period" of the channel, our scheme shows significantly lower session completion times. In particular, with 50% bad period on the channel, the default TCP session completion time is about 11 times more than the corresponding time with a perfect channel, while with our scheme, it is only 4 times larger. The source idle time is kept well under check and thus offers high performance benefit. Our scheme is suited for UMTS networks since we exploit: (a) the SDU discard function to inform TCP sources of packet losses during a bad wireless state and (b) the notion of flows available with PDP contexts.

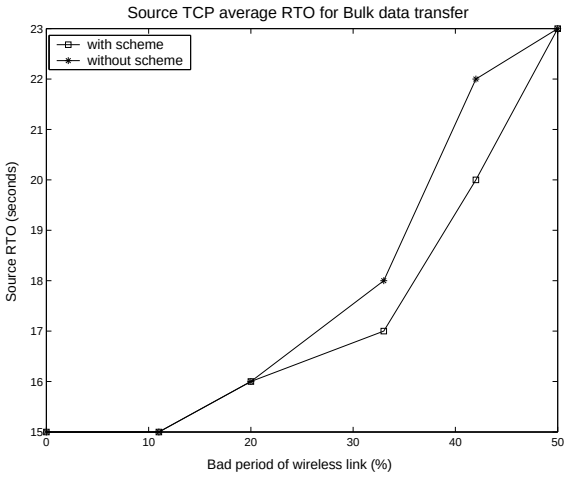

Fig. 7. *Average RTO (in seconds) for bulk traffic source.*

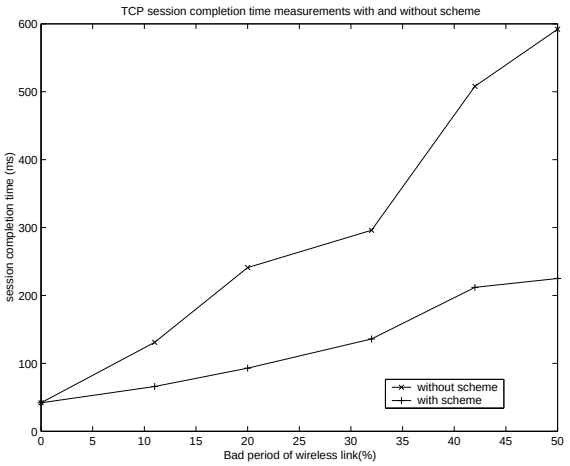

Fig. 8. *TCP Session time measurements with and without the scheme.*

VI. SUMMARY AND CONCLUSIONS

We have proposed an adaptive source timeout scheme for short flows suitable for wired cum wireless environments both for TCP short and bulk data transfer. Our scheme is based on explicit feedback (carried in ICMP messages) about the condition of the wireless link and encompasses both a premature source timeout and an extended timeout. The scheme is novel and simple since sources take appropriate decisions based on their own RTO time lines. Our results are very conservative - we have taken only small savings in timeouts and we have taken a fine-grained RTO timer with just 6 ARQ attempts. All the above parameters can be easily tuned to obtain even higher performance.

REFERENCES

[1] Hari Balakrishnan, Venkata N Padmanabhan, Srinivasan Seshan and Randy H Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *Proc. ACM SIGCOMM*, August 1996.

[2] W.K Jackson and Victor C.M.Leung, "Improving End-to End Performance of TCP Using Link-Layer Retransmissions over Mobile Internetworks", *Proc. IEEE ICC'99*, 324-328, 1999.

[3] Bikram S Bakshi, P Krishna, N.H. Vaidya, D K Pradhan, "Improving Performance of TCP over Wireless Networks", *Proc. of the* 17^{th} *International Conference on Distributed Computing Systems*, (ICDCS 1997).

[4] Durst et al, "TCP extensions for space communications", *Proc.* 2^{nd} *annual international conference on mobile computing and networking*, 1996.

[5] S.Goel and D. Sanghi, "Improving TCP performance over wireless links", *IEEE TENCON*, pp 332-5 vol.2, 1998.

[6] Gupta P K, Kuri J,"Reliable eln to enhance throughput of TCP over wireless links via TCP header checksum", *IEEE Globecom 02*, Vol 2, pp 1985 1989.

[7] A F Canton, T Chahed, End-to-end Reliability in UMTS: TCP over ARQ, *Globecom 01*, Vol6, pp 3473-3479.

[8] Neal Cardwell, Stefan Savage, Tom Anderson, "Modelling the Performance of Short TCP Connections", *Technical Report*, Computer Science Department, Washington University, Nov 1998.

[9] Marco Mellia, Ion Stoica, Hui Zhang, "TCP model for Short lived Flows", *IEEE Communications letters*, Vol6, No:2, February 2002.

[10] A. Feldman, J.Rexford, and R.Caceres, "Efficient policies for carrying Web traffic over flow-switched networks," *IEEE/ACM Trans. on Networking*, vol.6, pp 673-685, Dec 1998.

[11] Kostas Pentikousis, " TCP in Wired -Cum-Wireless Environments", *IEEE Communications Surveys*, Fourth Quarter 2000.

[12] Annex B: Test environments and deployment models. UMTS 30.3 v 3.2.0 TR 101 112 v3.2.0 (1998-04).

[13] Jaana Laiho, Achim Wacker and Tomas Novosad, "Radio network planning and optimisation for UMTS", John Wiley publications.