Performance of TCP Protocols in Deep Space Communication Networks

Özgür B. Akan, *Student Member, IEEE*, Jian Fang, and Ian F. Akyildiz, *Fellow, IEEE*

***Abstract*—The communication requirements for space missions necessitate to address the problems due to deep space communication networks. In this letter, the effects of slow start algorithm, propagation delay and the link errors on the throughput performance of transport layer protocols are investigated in deep space communication networks. The objective of this letter is to demonstrate through experimental results that existing TCP protocols are far from satisfying the deep space communication requirements and point out the urgent need for new TCP solutions.**

***Index Terms*—Congestion control, deep space communication links, high propagation delay, transport protocols.**

I. Introduction

THE space and outer planet exploration missions require advanced communication and networking technologies to be developed. The communication infrastructure will consist of interplanetary backbone network, planet proximity networks and on-board mission vehicles networks [1]. One of the most challenging parts of this infrastructure is the interplanetary backbone, which is composed of deep space communication links [2].

The deep space links are mainly characterized by link errors and extremely high propagation delays [3]. However, the currently existing TCP protocols have been designed to operate efficiently over the traditional almost error-free wired links with relatively low propagation delays ranging from milliseconds to a second. The window-based and acknowledgment-triggered congestion control mechanisms of existing TCP protocols suffer from high propagation delay and random link errors. Since the transmission rate is increased with round-trip time (RTT) granularity, an increase in RTT directly causes a decrease in the transmission rate acceleration and hence, in the overall link utilization. In the current literature, the congestion and rate control studies for links with high bandwidth-delay products and random loss mostly refer to Geo-stationary Earth Orbit (GEO) satellite links [4]–[6]. These links have RTT values around 550 ms which are still very low compared to deep space communication links. In order to assess the TCP performance over deep space links, we perform a wide range of simulation experiments and present the results in this letter.

Manuscript received June 19, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. S. Pierre. This work was supported by NASA-Glenn Research Center under the Grant NAG3-2585.

The authors are with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: akan@ece.gatech.edu; jfang@ece.gatech.edu; ian@ece.gatech.edu).

Digital Object Identifier 10.1109/LCOMM.2002.805549

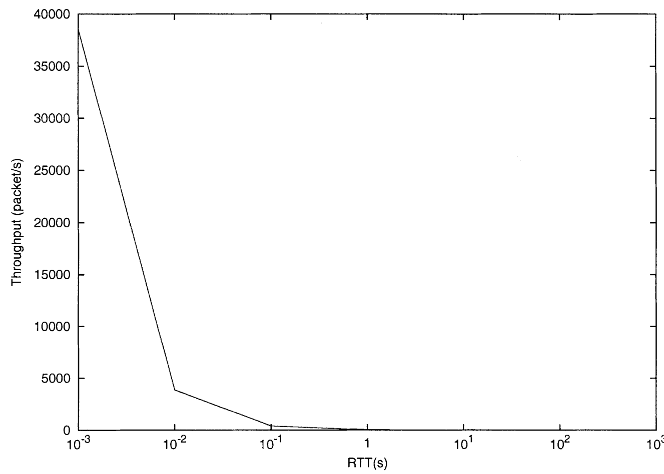

Fig. 1. Throughput for varying RTT from steady-state model.

II. Analytical Observations

The effect of RTT on the throughput can be inferred from the steady-state throughput model obtained for TCP-Reno in [7]. The asymptotic throughput equation as a function of packet loss probability p and RTT is given in [7] as

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

where T_0 is the initial retransmission time-out (RTO) value.

In Fig. 1, the throughput of an analytical TCP-Reno model is shown for packet loss probability of $p = 10^{-3}$ and modifying RTT values on a logarithmic scale starting from 10^{-3} to 10^{3} seconds. The throughput of a TCP connection is inversely proportional to RTT and decreases drastically for increasing RTT. At $RTT = 1000$ s, i.e., approximately 16.7 min, the achievable throughput degenerates to 308 b/s for a segment size of 1 KB. Note that $RTT = 16.7$ min is in the range of possible RTT values that can be experienced in deep space communications, e.g., between Mars and Earth.

III. Performance Evaluation Experiments

In order to explore the influence of the propagation delay on the performance of window-based and RTT-clocked transport protocols, we have performed simulation experiments with a wide set of TCP protocols. The experiments are performed with a very simple topology in order to isolate the effect of each factor under exploration. The source and destination are connected through 1 Mb/s link, which is assumed to have packet loss probability (P_{Loss}) of 10^{-3}. We first investigate the perfor-

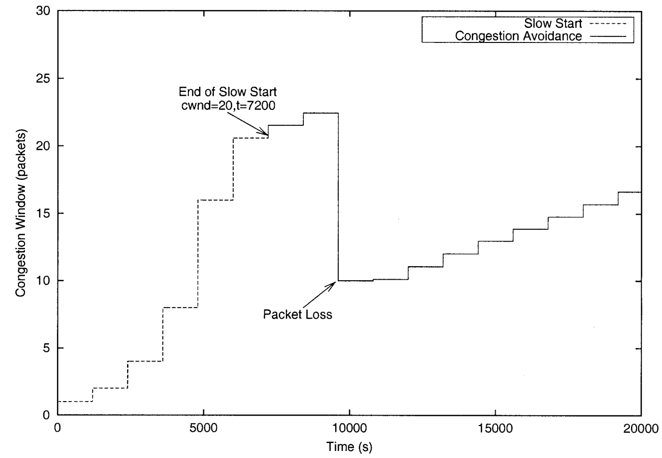

Fig. 2. Congestion window change of a TCP-Reno source during Slow Start and Congestion Avoidance.

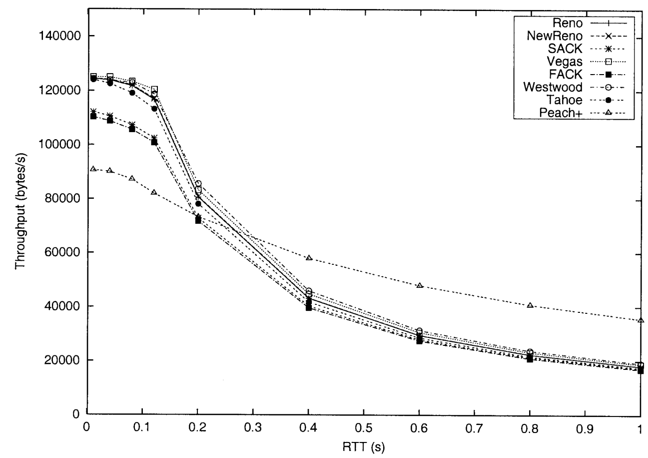

Fig. 3. Throughput performance of TCP implementations for low RTT ranges.

mance of TCP slow start algorithm. The effects of propagation delay and link errors on the performance will be presented afterwards.

A. Slow Start Performance

In the initial phase of the connection, TCP deploys the slow-start algorithm to probe the link availability and increase its congestion window ($cwnd$) size accordingly. The $cwnd$ is increased by one packet per new ACK received during this period for $cwnd < ssthresh$, where $ssthresh$ is the slow-start threshold window size. After congestion window reaches $ssthresh$, the slow start is terminated and the congestion avoidance algorithm is invoked. Although the slow-start algorithm is necessary for controlled window expansion in order to prevent congestion at the initial stages of the connection, it leads to poor throughput performance on high RTT links. The time interval, t_{SS}, that is underutilized during slow start can be expressed as follows:

$$t_{SS} = (\lceil \log_2 ssthresh \rceil + 1) \cdot RTT \qquad (2)$$

For instance, it takes approximately 2 hours for source to terminate the slow start period for $ssthresh = 20$ and $RTT = 40$ min.

To observe the effects of the slow start algorithm individually, we simulate TCP-Reno connection. Here, we assume $RTT = 20$ min, $P_{\text{Loss}} = 10^{-3}$ and $ssthresh = 20$. In Fig. 2, the congestion window evolution dependent on time is shown for the connection duration of 20 000 s. The source starts connection with the slow start phase at $t = 0$. At the end of $6 \cdot RTT$, the slow start is terminated and congestion avoidance is called at $t = 7200$ s. Hence, the link is not efficiently utilized for the slow start period of 120 min. This problem becomes much more severe for short duration connections on links with high propagation delays.

B. Propagation Delay

We perform simulations with the same topology and a wide set of TCP schemes including TCP-Tahoe, TCP-Reno, TCP-NewReno, TCP-Sack [8], TCP-Vegas [9], TCP-Fack [10], TCP-Westwood [11], and TCP-Peach+ [7], [13]. We observe the propagation delay effect for three different round-trip time ranges, i.e., low, high and very high as shown in Table I. In this table, very high RTT range covers RTT values representing deep space communication links.

TABLE I
RTT RANGES USED FOR THROUGHPUT PERFORMANCE EVALUATION

RTT Range	RTT_{Min}	RTT_{Max}
Low	0.001s	1s
High	1s	100s
Very High	100s	2400s

In Fig. 3, throughput values of all existing TCP protocols are around 120 KB/s at $RTT = 1$ ms. The throughput of all schemes start decreasing with increasing RTT values. For $RTT = 1$ s, the link utilization achieved by all transport protocols drop more than three times in magnitude. The major reason for performance decrease with increasing propagation delay is that TCP deploys window-based congestion control which increases its window size according to ACK reception for the transmitted packets at each RTT. The longer propagation delay is experienced, the slower window increase and hence lower throughput can be achieved. For low RTT values TCP-Westwood provides best performance among all TCP implementations. For 300 ms $\leq RTT \leq$ 1 s, TCP-Peach+ [7], [13] outperforms other schemes with throughput improvement of approximately 90%. This is mostly because of the *Jump Start* and *Quick Recovery* algorithms specifically designed for high bandwidth-delay product links [12].

The throughput achieved for all protocols keep decreasing drastically for high RTT ranges as shown in Fig. 4. The link utilization drops to approximately 5 KB/s for RTT values around 5 s. For high RTT values, TCP-Peach+ [7], [13] achieves significant throughput improvement compared to other protocols.

In Fig. 5, we show the throughput performance for very high RTT values including the RTT range for communication links between Mars and Earth, i.e., 9 to 50 min based on the orbital location of planets. In this case, the throughput achieved by TCP protocols drops down to approximately 10 B/s. This decrease is very drastic and the entire link remains almost unutilized. Although TCP-Peach+ outperforms other TCP schemes for high RTT values, the performance degradation is yet too serious that it can only achieve 20 B/s link utilization for 40 min round-trip time.

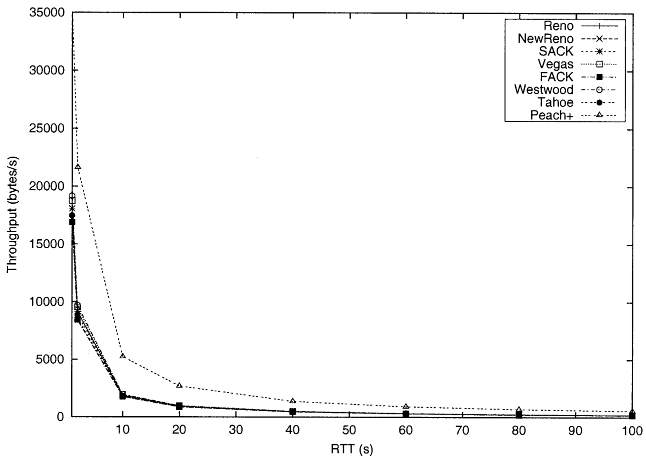

Fig. 4. Throughput performance of TCP implementations for high RTT ranges.

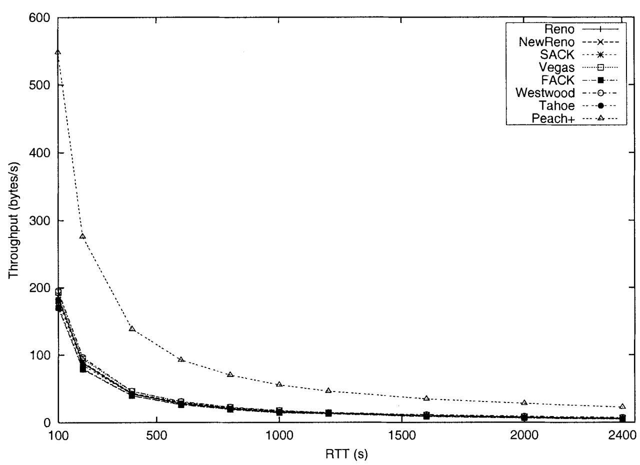

Fig. 5. Throughput performance of TCP implementations for very high RTT ranges representing deep space communication links.

C. Link Errors

The recognition of a packet corruption due to space link error as congestion loss leads to unnecessary window decrease and hence severe throughput degradation [13]. This problem is amplified with extremely high propagation delay, since it takes longer time for window-based mechanisms to recover from unnecessary congestion window throttle. In Fig. 2, a packet is lost due to link error at $t = 9500$ s. It takes approximately $11 \cdot \mathrm{RTT}$ for TCP source to resume its original rate before halving.

In order to show the effect of link errors on throughput performance, we perform simulations with same topology and $RTT = 20$ min for varying packet loss probability (P_{Loss}). In Fig. 6, the results show that increase in packet loss probability decreases the throughput achieved by all protocols as expected. TCP-Peach+ achieves approximately 246% throughput improvement. However, for even very small loss probability of $P_{\mathrm{Loss}} = 10^{-6}$, the throughput achieved by TCP-Peach+ is only 52 B/s and other TCP schemes is around 15 B/s. Therefore, the high propagation delay is dominant challenge for throughput performance in deep space links.

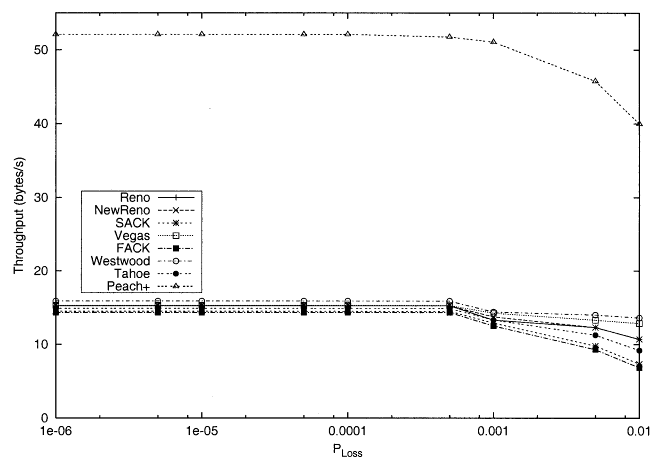

Fig. 6. TCP throughput performance for varying P_{Loss} with $\mathrm{RTT} = 20$ min.

IV. Conclusion

In this letter, we have presented simulation experimental results for TCP performance on deep space links. It is shown that window-based legacy transport protocols provide very poor performance on deep space links with extremely high propagation delay and random link errors. The high propagation delay is the most important challenge to be addressed in order to meet communication requirements for deep space missions.

References

[1] K. Bhasin and J. Hayden, "Space internet architectures and technologies for NASA enterprises," in *IEEE Aerospace Conf.*, 2001.

[2] "Space Communications Protocol Specifications (SCPS)—Rationale, requirements and application notes," Consultative Committee for Space Data Systems, CCSDS 710.0-G-0.3, 1997.

[3] R. Durst, G. Miller, and E. Travis, "TCP extensions for space communications," in *Proc. ACM Mobicom*, Nov. 1996, pp. 15–26.

[4] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. Networking*, vol. 5, pp. 336–350, June 1997.

[5] J. Tang, G. Morabito, I. F. Akyildiz, and M. Johnson, "RCS: A rate control scheme for real-time traffic in networks with high bandwidth-delay products and high bit error rates," in *Proc. IEEE INFOCOM 2001*, vol. 1, 2001, pp. 114–122.

[6] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite IP networks," *IEEE/ACM Trans. Networking*, pp. 307–321, June 2001.

[7] J. Padhye, V. Firoio, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, pp. 133–145, Apr. 2000.

[8] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanov, "TCP selective acknowledgment options,", RFC 2018, 1996.

[9] L. S. Brakmo, S. O. Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proc. ACM SIGCOMM*, Oct. 1994, pp. 24–35.

[10] M. Mathis and J. Mahdavi, "Forward acknowledgment: Refining TCP congestion control," in *Proc. ACM SIGCOMM*, Aug. 1996, pp. 281–292.

[11] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. ACM Mobicom 2001*, Rome, Italy, July 16–21, 2001, pp. 287–297.

[12] I. F. Akyildiz, X. Zhang, and J. Fang, "TCP-Peach+: Enhancement of TCP-Peach for satellite IP networks," *IEEE Commun. Lett.*, vol. 6, pp. 303–305, July 2002.

[13] E. Criscuolo, K. Hogie, and R. Parise, "Transport protocols and applications for internet use in space," *Proc. IEEE Aerospace Conf.*, vol. 2, pp. 951–962, Mar. 2001.