

Delay Resistant Transport Protocol for Deep Space Communication

Mohanchur Sarkar¹, Kaushal K. Shukla², Kankar S. Dasgupta¹

¹*SATCOM and Navigation Applications Area, Space Applications Centre (ISRO), Ahmedabad, India*

²*Department of Computer Engineering, Institute of Technology, Banaras Hindu University, Varanasi, India*

E-mail: mohanchur@yahoo.com, {msarkar, ksd}@sac.isro.gov.in, kkshukla.cse@itbhu.ac.in

Received November 30, 2010; revised January 12, 2011; accepted January 14, 2011

Abstract

The throughput of conventional transport protocols suffers significant degradation with the increased Round Trip Time (RTT) typically seen in deep space communication. This paper proposes a Delay Resistant Transport Protocol (DR-TCP) for point-to-point communication in deep space exploration missions. The issues related to deep space communication protocol design and the areas where modifications are necessary are investigated, and a protocol is designed that can provide good throughput to the applications using a deep space link. The proposed protocol uses a cross layer based approach to find the allocated bandwidth and avoids initial bandwidth estimation. A novel timeout algorithm estimates the timeout duration with an objective to maximize throughput and avoid spurious timeout events. The protocol is evaluated through extensive simulations in ns2 considering high RTT values typically seen in Lunar and Mars Exploration Networks under different conditions of packet error rates. DR-TCP provides a significant increase in the throughput as compared to traditional transport protocols under the same conditions. A novel adaptive redundant retransmission algorithm is also presented to take care of the high PER in deep space links. The effect of the Retransmission Frequency has been critically analyzed considering both Lunar and Deep Space scenarios under different levels of PER. The results are very encouraging even in high error conditions. The protocol exhibits a RTT independent behavior in throughput, which is the most desirable quality of a protocol for deep space communication.

Keywords: Delay Resistant, TCP SACK, TCP Vegas, Deep Space Communication Protocols, RTT, TP-Planet, Saratoga, LTP, DS-TP

1. Introduction

Conventional Transport Protocols show significant performance degradation when used over long propagation delay links. Channel errors worsen the performance as has been critically analyzed in [1]. The design of a transport protocol for deep space communication remains a challenge for protocol developers mainly to provide a sustainable throughput when the RTT increases from milliseconds to seconds to even minutes as seen in deep space links [2,3]. All the major transport protocols produce throughput in the order of bytes/sec when used with RTTs typically seen in Mars exploration missions [1]. It has also been emphasized that high delay remains the major challenge for deep space protocols as compared to other contributing factors like channel error, link disruption and bandwidth asymmetry because of the limit im-

posed by the speed of light [1].

Thus, there is a need to develop a transport protocol whose performance will be independent of the RTT to make it a candidate for deep space communication [4]. The second most important detrimental factor for deep space communication is the high bit error rate [5], which translates into a high packet error rate and makes any ARQ based protocol unable to provide reliability of the transmitted packets [6]. The third most important point is the link disruption problem which has to be handled by the protocol and fourth being the asymmetry in forward and return bandwidth capacity [4,5,7].

In this paper, we present a new transport protocol, which is Delay-Resistant, Error-Resilient, and always provides a very good throughput under increasing RTT of the link and varying channel errors.

Delay Tolerant Networks (DTN) provide a generic

solution to the problems of deep space communication [8,9]. They provide a new architecture of the way data transfer should happen in deep space networks as a series of store and forward methods of communication mainly to handle the problems imposed for the Interplanetary Internet [3]. The DTN unifies the different regional networks by the use of the bundle layer whose specification is provided by DTNRG [9]. Although, different DTN Convergence Transport Protocols have to be designed to suit the environment under consideration. The DTN [8] acts as a framework within which different transport protocols suitable for the link coexist, and are glued to the overall operation of the network by the Bundle Layer [8]. This has given rise to the different DTN Convergence Transport Protocols like Saratoga [10], LTP [11-13] and DS-TP [5] to be used in DTNs.

TP-Planet [7] is also a well formed transport protocol for deep space communication and provides a generic solution for the development of a full fledged network beyond the present requirement [5,10]. All these solutions are generic but they call for the development of a new protocol stack or a major modification in the working of the protocol and demand special support from the network [9]. The applications developed for this type of network should also be specially tailored and be different from applications used in conventional networks.

In this paper, we provide a transport protocol, which is a modification of the conventional Transport Protocols and does not need any change as far as the packet structure of the TCP/IP protocol stack is concerned.

It has been designed keeping in mind of a situation where there is a point-to-point communication requirement. Consider a scenario where a spacecraft is sent exclusively for Lunar or Mars exploration as described in the Mars Near-Term Communication Architecture in [14], where communication is done directly from the spacecraft to any of the DSN stations located at different parts of Earth. In this type of exploration missions, communication between the transmitter and the receiver is a point-to-point connectivity with the major problem being the huge propagation delay and high packet error rates of the channel. The time during which the link will be available in this case can be predicted from the available ephemeris data of the orbiting spacecraft, so the predicted link disruptions can be handled by an understanding of the orbital dynamics.

The Delay Resistant TCP is specially designed to work on this type of exploration missions. It can also be used as a DTN convergence Layer protocol in the transport protocol to be used in the IPN Backbone links [3,15], as they are also point to point and have similar communication constraints of high delay and error [1,3,9].

2. Causes of RTT Related Degradation

As already discussed it has been found that conventional TCPs do not perform well with increased RTT and become completely unusable when used for interplanetary distances [1,3,9]. In this section, an attempt has been made to investigate the causes of RTT dependent performance degradation for conventional TCP protocols.

2.1. Initial Bandwidth Estimation

All transport protocols go for slow start phase to have an initial estimation of the bandwidth availability in the network. This is a well-proven and justified technique in a network shared by multiple connections and with small RTT. When RTT starts increasing the bandwidth delay product, which signifies the amount of data needed to fully utilize the channel, also increases and the time needed by slow start to attain a sufficient level of data transmission becomes high. Thus, a majority of the time is spent in finding the available bandwidth and then start utilizing it. This creates a severe underutilization of performance for conventional transport protocols.

2.2. Reduced AIMD Parameters

After the slow start threshold is reached, Transport protocols move to the congestion avoidance phase where generally, the additive increase parameter is one and multiplicative decrease parameter set to 1/2. This is a very good way to conservatively increase the transmission rate in case of networks, where RTT is very less with many connections sharing the bandwidth and where the chances of getting into congestion state is very high [16]. However, in the case of deep space communication, where the RTT is in the order of secs or even minutes, a low additive increase parameter seriously affects the performance of the protocol as considerable time is spent in attaining the actual capacity of the network. Moreover, the Multiplicative Decrease Parameter of 1/2 is very well justified [17] in terrestrial network being an effective means in providing the fairness property [18] of the protocol and handling congestion, but creates a huge degradation as the congestion window size for deep space communication is quite large to fully utilize the channel. Therefore, once a multiplicative decrease is applied, it may take a long time for the protocol to regain the value from where it fell.

2.3. Handling of Errors as Congestion

TCP being developed for wired connectivity always assumes any packet loss as a sign of congestion [17]. Therefore, whenever a packet loss is signaled by a DUP

ACK or timeout, the multiplicative decrease parameter is applied and the congestion window is generally reduced to half of its prevailing value. If the reduced *cwnd* is less than the slow start threshold, the slow start process is applied; otherwise, congestion avoidance slowly increases the *cwnd*. In case of wireless communication, the major factor of packet loss is because of channel error but transport protocols not being able to differentiate, drastically reduces the throughput—a very common problem for all wireless communications with the degradation being amplified in the deep space scenario.

2.4. Timeout Mechanism

The inability of the Retransmission Timeout algorithm in accurately predicting the time of waiting for ACKs plays a predominant role in the degradation of throughput for conventional transport protocols. The binary backoff mechanism though being highly appropriate in terrestrial Internet for preventing congestion creates huge degradation when applied in large RTT [17] and in cases where congestion is not the reason for timeout. The initial value of the timeout is generally kept to 3 secs in many TCP variants and can at most be backed off 64 times to 192 secs so any RTT larger than this becomes unreachable. This is termed as the Protocol Radius [19] beyond which timers prevent communication. Timeout is the most expensive event that any TCP protocol may encounter as with every timeout the congestion window is reduced to one and the slow start threshold reduced to operate with a very conservative approach. All packets after the point of highest acknowledged data are sent again in the event of a timeout. This creates severe degradation to the congestion window, as timeouts are very common in deep space links prone to packet error.

3. Design Consideration of the Protocol

Considering the pre scheduled nature of the planetary exploration missions at present [5,10,12], for a point to point deep space communication the bandwidth available to the sender can be known from an investigation in the Data Link layer which directly connects with the physical layer for data transmission. The proposed Delay Resistant TCP is composed of two main algorithms: 1) Cross Layer Bandwidth Estimation, and 2) Transmission Maintenance Procedure.

3.1. Cross Layer Bandwidth Estimation

During this process, the bandwidth of the channel allocated is obtained from a cross layer reference to the data link layer. The ideal RTT or the propagation delay of the

spacecraft and the Earth station can be obtained from the ephemeris data available in the network layer. From the bandwidth allocated to the sender and the Ideal RTT the Bandwidth Delay Product (BDP) in terms of packets is obtained for the link concerned as given in (1)

$$BDP = \frac{Bandwidth(bits/s) \times Ideal_RTT(sec\ s)}{Packet_length(bits)} \quad (1)$$

This gives the value of the congestion window that will keep the total pipe between the transmitter and receiver full. In one RTT if BDP numbers of packets are transmitted, the link will be fully utilized, as the sender will go on sending the packets, which will take one full RTT to finish. Immediately after the transmission of the last packet as per BDP computation, the ACK for the first packet transmitted in that round will be received. This will ensure that the sender spends no idle time in waiting for the ACKs to determine the next congestion window. This is a very important consideration for deep space communication to ensure that the sender is always in a transmitting mode and effectively using the available channel capacity. In all the window-based conventional transport protocols with an ACK clocked transmission, the sender remains idle most of the time. Hence, if the sender can be kept busy all the time in transmitting packets the adverse effect of RTT degrading the performance can be done with and a major improvement can be achieved in the performance.

In this paper, the results have been obtained assuming that the ideal RTT is available from an ephemeris calculation module and half of the BDP calculated from that is used in the first RTT. This is to ensure that there would not be an overestimation of the channel capacity in the first RTT itself. Using the initial bandwidth estimation technique the time wasted in slow start method used in conventional TCP can be avoided. This is quite logical in the context in which the communication is sought, where the share of bandwidth is already known prior to the communication starts.

3.2. Transmission Maintenance Procedure

After one RTT once more accurate BDP is derived, the protocol moves to the Transmission Maintenance Procedure where the congestion window is kept at the value of the BDP. There is no AIMD parameter associated, because this value of the congestion window is the optimal value required for full utilization of the channel capacity.

One of the main reasons for conventional TCP protocol throughput degradation is the modification of the congestion window with AI parameters depending on the arrival of the ACK. The arrival of the ACK is dependent on the RTT and so transitively the value of the conges-

tion window or rate of transmission becomes coupled with the RTT.

In DR-TCP, the transmission rate is decoupled from the ACK reception, which in turn makes the protocol independent of RTT. In DR-TCP, the ACKS are only used to ascertain reliable delivery of packets. The sending rate is determined and maintained as obtained from the Cross-Layer Bandwidth Estimation procedure.

DR-TCP uses Selective Acknowledgement method to signal the missing packets, as SACK [20] is very effective in handling multiple losses in the same window. Inside the Transmission Maintenance Procedure whenever three DUP ACK or a SACK block is received the protocol just Fast Retransmits the lost packets. No reduction is done to the congestion window by the Multiplicative Decrease factor. This is because of the fact that DR-TCP assumes that any loss of packet is because of channel noise and not due to congestion. So decreasing the congestion window is not going to serve any purpose.

A significant feature of DR-TCP is that though it does not perform the congestion avoidance procedure like conventional TCP, the congestion control is implicitly handled by always keeping the congestion window equal to the value of BDP. Deep space links are highly prone to wireless channel errors so timeouts are likely to happen not because of an under valued timeout duration but due to packets being repeatedly corrupted in the channel [17]. Therefore, in DR-TCP for the case of timeout there is no reduction of congestion window only a retransmit of lost packet is done.

Every RTT, before the starting of a round, the Ideal RTT value is checked as predicted from the ephemeris calculation module. The value of the BDP is likewise modified to suit the prevailing propagation delay. This is necessary because if the ideal RTT or the propagation delay decreases due to a decrease in distance, the DR-TCP sender should not be transmitting more packets than the capacity and should not create a situation in which congestive loss happens due to buffer overflow. The overall operation of the protocol is depicted in the flow chart in **Figure 1**.

3.3. Retransmission Timeout Algorithm

The DR-TCP uses a novel timeout algorithm designed with the concept of normalized mean rise values MR_i proposed by authors [17] as follows.

$$MR_i = \frac{Mi - IRTT}{IRT T} \quad (1)$$

where Mi is an integral mean of the mean RTT values for all the windows starting from the inception of the connection and $IRT T$ is the latest ideal RTT value available from the ephemeris calculation module. MR_i is a

very good estimate of how much extra delay, including the queuing delay and the processing delay, exists on top of the propagation delay between the two communicating entities.

If T_E denotes the timeout estimated by the algorithm and T_A the actual time when the ACK returns then T_{OE} gives the amount of extra time the sender waits for the ACK to return.

$$T_{OE} = T_E - T_A \quad (2)$$

A very small value of T_{OE} will lead to spurious timeouts and a large value will lead to underutilization of channel capacity. The best performance of the protocol happens when T_{OE} is very less but still does not create timeouts, which is only possible when estimated arrivals are very well predicted by actual arrival of the ACKs. This is only possible when there is very less variance in the arrival of the ACKs. In case of DR-TCP the traffic pattern is mostly going to be of the type of large file transfers as any recorded images or sensor data in the form of files will be sent back to Earth [11,10,18]. Even a video will also be sent after being stored as a file transfer. Thus, the traffic pattern is not expected to be busy as seen in the terrestrial Internet. In this scenario timeout can be very accurately estimated as in (3)

$$T_E = (1 + MR_i) \times IRT T \quad (3)$$

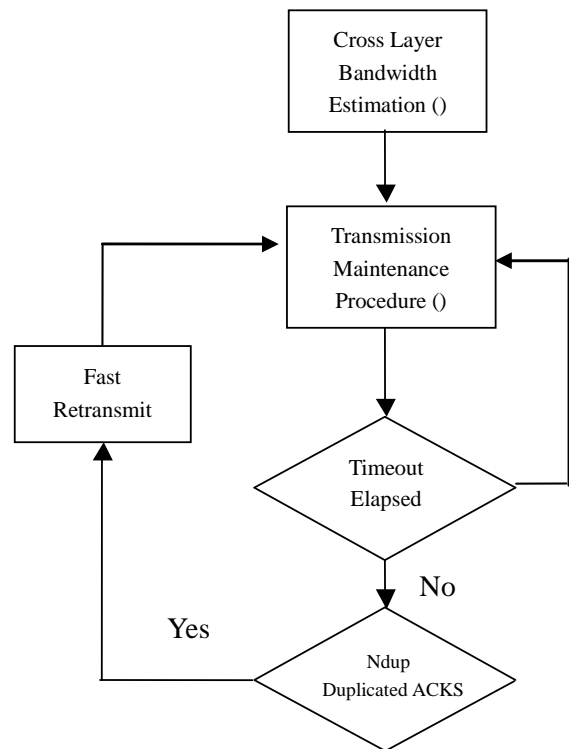


Figure 1. Flow chart of DR_TCP.

where IRTT is the latest value of Ideal RTT received from the ephemeris calculation module and MRi denotes the best estimate of how much extra delay is generally seen in the system [17].

Other issues related to the use of timeout algorithm are the initial value of the timeout, as it takes few RTTs to have a proper value of the normalized mean. This is very important because from simulation experiments it has been found that the timeout algorithms generally fails during the start of the connection when proper estimates of the averaged RTT values and variances is not available.

DR-TCP does not use the binary back off algorithm, which tends to increase the RTT to very high values. Instead, the initial value of the timeout is kept equal to $2 * \text{IRTT}$ value. This is to keep enough safety margins to prevent timeout at the start of a connection. The MAXRTO or the maximum value of the timeout is set to $2 * \text{IRTT}$. This is because in the case of a point-to-point communication the major factor for a timeout is the high packet error and in that case increasing the timeout is not going to solve the problem.

4. TEST and Evaluation

The DR-TCP was implemented in ns2 [21] and simulations were performed with one node acting as transmitter communicating to another node acting as the receiver with RTT which is much more than the terrestrial case.

4.1. Lunar Exploration Network

We have considered the Lunar Exploration Network with the parameters given in **Table 1**. Since the throughput of any transport protocol depends on the size of the file to be transferred and generally the throughput is seen to increase with increasing file size [5,7], here we have considered the case where the sender goes in for a continuous FTP transfer to the receiver throughout the simulation time. The throughput obtained in this simulation experiment is independent of file size selected for transmission. The main aim of the simulation was to see the performance of the protocol with an increased RTT value of 2.560 secs. The simulation was performed with different packet error rates in the channel to see the behavior of the protocol under different error conditions.

To compare the performance of the protocol with other conventional transport protocols, TCP SACK [20] and TCP Vegas [22] were also simulated as transport protocol under similar conditions. The results are shown in **Table 2**.

In the simulation experiments, as shown in **Figure 2** that DR-TCP provides good throughput under various

conditions of packet error rates. To see the impact of mainly RTT, on the throughput, a condition with very low PER of 10^{-6} was considered in the simulation. In this case, the throughput achieved is only dependent on RTT. It can be seen from **Table 3**, that under PER of 10^{-6} for a lunar case all the protocols perform quite well. The performance degradation starts with increasing Packet Error Rates. At PER of 10^{-5} the throughput of all the protocols is quite good. From this result, it can be concluded that if the packet error rate can be kept at a low value, substantial bandwidth utilization is possible for the lunar case. Degradation of TCP SACK and Vegas starts with PER at 10^{-3} . At PER of 10^{-1} SACK and Vegas give very little utilization of throughput. This is because of the fact that as error increases in the channel, if the protocol is not able to differentiate between congestion and errors the performance degrades. DR-TCP is comparatively seen to produce better results even in case of 10^{-2} and 10^{-1} error cases with 59% and 32% throughput respectively in **Table 3**.

From these experiments, it can be concluded that for the lunar distance, RTT is not the major factor of throughput degradation, which can be seen in the very low PER cases. The combination of high RTT with error in the channel actually makes the performance degradation very abrupt. Thus, in the LUNAR case the accuracy of the algorithms by which differentiation of packet loss due to congestion and corruption is important for performance enhancement.

4.2. Mars Exploration Network

Similar to the LUNAR Network the protocol was simulated with same parameters as **Table 1** but increased RTT and BDP to match the mean RTT seen in the case of Mars Exploration of 26.21 mins. Simulation was run for 1000 times the RTT value and from **Table 4** it can be seen that in the case of RTT of 26.21 mins the throughput of TCP SACK and TCP Vegas has drastically reduced to almost zero even in the case when PER is 10^{-6} . This shows the extent to which RTT has a devastating effect on the performance. From **Table 5** it can be seen that in this case the Packet Error Rate does not matter, as the throughput itself is so low because of increased RTT. Under a very low PER of 10^{-6} it can be seen that DR-TCP gives 90.99% utilization of the bandwidth as compared TCP SACK and Vegas. This simulation experiment shows the advantages that DR-TCP has because of the modified design of the protocol.

Moreover, it can be seen that in the case of DR-TCP there is a slow degradation of throughput with increasing PER. In the case of moderate PER values, DR-TCP provides good throughput of 65% utilization in case of 10^{-5}

PER and a 52% utilization for 10^{-4} PER. Only when the PER drops below 10^{-3} the utilization reduces drastically to 22% down to 8% in case of PER 10^{-1} .

4.3. Deep Space Network

The simulation with an RTT of 10 mins is to show the performance of the protocol for the case when the MARS is nearer to earth and as a theoretical evaluation of the performance for mid level RTT. This work is done also for a comparison of DR-TCP with TP-Planet [7] and ARC [23]. Both TP-Planet and ARC published their protocol performance considering 10 mins RTT so a comparison can be made with these deep space protocols.

In **Table 6**, for the case with very low PER of 10^{-6} , it can be seen that DR-TCP gives a 94.997% utilization of the channel. This shows capability of the protocol to handle high RTT values. For a link with RTT = 600 secs and PER = 10^{-5} , TP-Planet gives a 66% utilization, as compared to DR-TCP giving 79.281% utilization. For a comparison with ARC [23] considering an RTT = 600 secs at PER of 10^{-4} , ARC produces approximately 10% utilization, the performance dropping drastically to a 1% utilization for PER of 10^{-2} . In comparison to this DR-TCP obtains 56.742% utilization at 10^{-4} and 26.585% utilization for PER of 10^{-2} .

4.4. RTT Independence Characteristics

From **Table 7** it can be seen that the throughput of the DR-TCP degrades very slowly with a huge increase in RTT. **Table 8** shows the percentage utilization and it can be seen that the utilization degrades very slowly when compared with a RTT of 2.5secs to a RTT of 26.21mins case, considering very low PER of 10^{-6} . This is the most important characteristic of the protocol, *i.e.*, the RTT independent behavior, which has been the main aim in the design of the protocol. This Delay Resistant behavior makes the protocol a good candidate for deep space communication.

Here, we have not considered the FEC techniques, which need to be applied in the physical layers to provide a good BER of the channel. The BER obtained is dependent on many parameters related to the core RF communication. These include choice of modulation technique, transmission and reception efficiency of the transmitter, and the receiver concerned, the distance at which they communicate, the size of antennas etc. Therefore, we have not considered those parameters as they vary from the way missions are designed to operate [24,25]. Here, the term bandwidth means the actual bandwidth available to the transmitter after the applica-

tion of FEC so that our analysis of the transport protocol can be done independently of the way the RF communication is planned.

Table 1. Lunar network simulation parameters.

Distance	384000 km
Ideal RTT	2.560 secs
Bandwidth	500 kbps
Packet Size	1000 bytes
BDP	153 packets
Cwnd	153 packets
Simulation Time	2560 secs

Table 2. Throughput of TCP variants for lunar distance RTT = 2560 ms.

PER	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
TCP SACK (kbps)	490.272	471.02	365.46	130.55	37.35	8.32
TCP Vegas (kbps)	468.130	469.54	420.23	184.56	52.43	7.56
DR-TCP (kbps)	493.246	492.924	486.882	437.037	299.780	160.358

Table 3. Percentage utilization of TCP variants for lunar distance RTT = 2560 ms.

PER	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
TCP SACK (%)	98.054	94.205	73.093	26.111	7.4716	1.6659
TCP Vegas (%)	93.326	93.908	84.046	36.913	10.487	1.5122
DR-TCP (%)	98.649	98.584	97.376	87.407	59.956	32.071

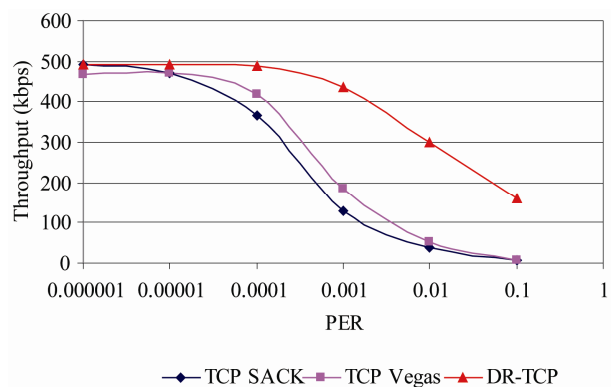


Figure 2. Throughput of DR-TCP for lunar RTT.

Table 4. Throughput (kbps) of TCP variants for mars distance RTT = 26.21 mins.

PER	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
SACK	0.010564	0.010564	0.010564	0.010538	0.010512	0.009856
Vegas	0.010586	0.010586	0.010586	0.010586	0.010586	0.010443
DR-TCP	454.995	325.525	264.19	221.684	113.179	40.986

Table 5. Percentage utilization of TCP variants for mars distance RTT = 26.21 mins

PER	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-3}	10^{-1}
SACK (%)	0.0021128	0.002113	0.002113	0.002108	0.002102	0.001971
Vegas (%)	0.0021172	0.002117	0.002117	0.002117	0.002117	0.002089
DR-TCP (%)	90.999	65.105	52.838	44.3368	22.6358	8.1972

Table 6. Percentage utilization of DR-TCP for RTT = 10 mins.

PER	Utilization (%)
10^{-6}	94.997
10^{-5}	79.281
10^{-4}	56.742
10^{-3}	50.012
10^{-2}	26.585
10^{-1}	8.6074

Table 7. Throughput of DR-TCP for varying RTT at PER = 10^{-6} .

RTT	2.560 secs	10 mins	26.21 mins
(kbps)	493.246	474.985	454.995

Table 8. Utilization of DR-TCP varying RTT at PER = 10^{-6} .

RTT	2.560 secs	10 mins	26.21 mins
%	98.649	94.997	90.999

4.5. Observation

From all the simulation experiments, we arrive at general inferences that DR-TCP is indeed RTT independent and provides a very good throughput in high RTT ranges. For conventional Transport Protocols delay is not the major problem of performance degradation in case of Lunar type RTT. A combination of high delay and high packet error rate degrades performance so emphasis has to be given to handle errors in the channel. On the other hand, for Mars Exploration, the RTT degradation is most predominant and the reduction is of such a higher magnitude, that the difference of increasing packet error rates hardly

matters. In that case, first the degradation due to large RTT has to be handled and then the degradation arising from channel errors.

In the design of DR-TCP, this philosophy is adopted and it can be seen that DR-TCP has increased the performance of the protocol even under very large RTT values. The performance is very good under low error conditions. The degradation in performance occurs only when the PER becomes very high. Thus, the issue of delay has been handled effectively, only there is a need to take care of the degradation under very high error conditions.

In the following section, this issue has been analyzed and a novel solution provided considering the different RTT cases.

5. Adaptive Redundant Retransmission Technique

In this technique, the lost packets are transmitted adding redundancy so that the probability of the retransmitted packets getting lost is reduced. In this context, we can see the way the other Deep Space Communication Protocols like Saratoga [10] and DS-TP [5] have handled the problem of retransmission.

Saratoga [10] uses a hole-filling algorithm where the transmission of packets happen in rounds with lost packets being retransmitted after the end of each round called a hole filling mechanism. The problem with this approach is that the number of rounds needed to complete the delivery is increased. Since the RTT is very high the application at the receiver gets the data after a long time, particularly when the error rate is high.

DS-TP [5] uses the Double Automatic Retransmission (DAR) approach where all packets are repeated after a fixed interval, which is known to the receiver in advance. This is to reduce the packet error probability at the cost

of redundancy added to each packet. In this approach since all packets are sent with redundancy the bandwidth is not used optimally, especially when the PER is less.

In DR-TCP, a new retransmission technique is adopted where redundancy is not added to the packets transmitted for the first time. Only the packets that are signaled as lost by the SACK blocks are transmitted with added redundancy. The logic for this approach is that if redundancy is added to all the packets there will be wastage in the bandwidth when the packet error rate is less. Moreover, considering the powerful FEC codes available, generally the system is designed to work at a fairly good BER [24,25]. In that case adding redundancy to all packets may not be that advantageous. On the contrary, if we add redundancy to lost packets the extra bandwidth wastage happens only for the lost packets and the protocol tries to ascertain that a packet once retransmitted should not get lost again as this type of repeated loss actually degrades the performance to the protocol to abnormally low levels. This also creates more timeout events as has been analyzed in Section 6.

In the following section, the Adaptive Redundant Retransmission Technique when applied on DR_TCP has been analyzed through simulation experiments. Here, for all the three cases considered in Section 4, the algorithm is applied and the performance improvements with different values of the Retransmission Frequency (RF) have been analyzed.

5.1. Lunar with More Redundant Transmission

The same experiment as in Subsection 4.1 for the Lunar Exploration Network was repeated using the increasing value of Retransmission Frequency (RF), which denotes the number of times the packet is repeated during retransmissions. In **Table 9**, the first column denotes the performance achieved when normal DR-TCP is used without the redundant retransmission technique. The Retransmission Frequency is doubled in the following columns to show its impact on the performance.

From **Figure 3**, it can be seen that under low error conditions the performance of the protocol remains same. This is because the redundant retransmission technique comes into play only when errors occur in the channel. The extra packets are not transmitted when PER is less as in seen from **Table 9**. Under PER 10^{-2} the performance improvement is not there. In case of PER 10^{-1} the utilization increases from 32% to 49% for a retransmission frequency of 4. With increased RF of 8 and 16, a slow degradation of throughput results because of increased number of redundant packets.

From the experiments, it can be concluded that for a

LUNAR case Retransmission Frequency (RF) of 4 is sufficient for a 50% utilization of the channel even under very high error rates.

5.2. Deep Space Network

The same simulation parameters as used in Subsection 4.3 have been used in this case with increased Retransmission Frequency to see the improvement in throughput. From **Figure 4** it can be seen that the throughput increases with increasing RF above PER of 10^{-3} . For PER of 10^{-2} , Retransmission Frequency of 2 gives a steady throughput of nearly 48%. In case of very high PER of 10^{-1} , it is seen that even RF of 4 is not sufficient to enhance the throughput and with RF of 8 only steady throughput of 47% is obtained. Increasing beyond RF = 8 decreases the performance because of more redundant packets being transmitted, especially in the case of PER = 10^{-1} where more retransmissions are likely.

From **Table 10** it can also be noticed that for errors lower than PER of 10^{-3} there is very less improvement. For PER more than 10^{-2} it gives significant improvement. In the very high error case of 10^{-1} , it can be seen that the improvement is highest with RF of 8. From the experiments, it can be concluded that with RF = 8 even at PER 10^{-1} and high RTT of 10mins there can be an appreciable utilization of the capacity of almost 50%.

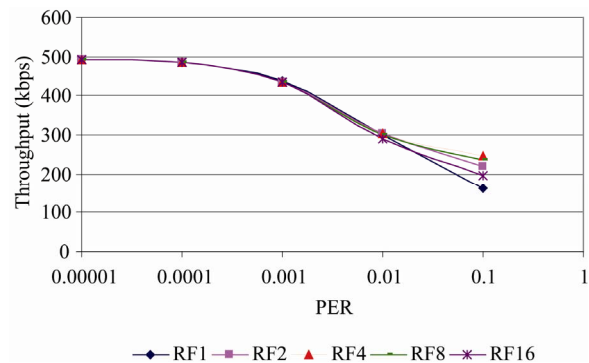


Figure 3. Throughput of DR-TCP LUNAR network.

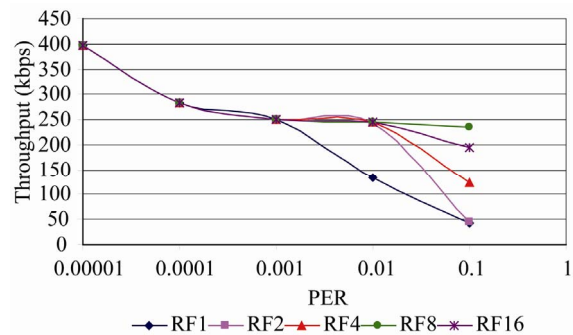


Figure 4. Throughput of DR-TCP for RTT = 10 mins.

Table 9. Percentage bandwidth utilization of DR-TCP with increasing retransmission frequency for RTT = 2.560 s.

PER	RF1	RF2	RF4	RF8	RF16
10^{-5}	98.5848	98.549	98.549	98.549	98.549
10^{-4}	97.3764	97.3394	97.3394	97.3394	97.338
10^{-3}	87.4074	87.2742	87.2716	87.2696	87.237
10^{-2}	59.956	60.5812	60.8146	60.24	57.9246
10^{-1}	32.0716	43.8586	49.0958	47.1112	38.7854

Table 10. Percentage bandwidth utilization of DR-TCP with increasing retransmission frequency for RTT = 10 mins.

PER	RF1	RF2	RF4	RF8	RF16
10^{-5}	79.281	79.2804	79.278	79.2764	79.2714
10^{-4}	56.742	56.8248	56.8144	56.7932	56.751
10^{-3}	50.0126	50.0608	50.0804	50.0008	50.0498
10^{-2}	26.5856	48.4458	49.1642	49.167	49.1036
10^{-1}	8.6074	9.0888	25.0222	47.0166	39.0084

5.3. Mars Exploration Network

For RTT of 26.21 minutes, the scenario for Mars Exploration with increasing RF values given in **Table 11** and **Figure 5**, it can be seen that the throughput increases with increasing RF above PER of 10^{-3} . For PER of 10^{-2} , Retransmission Frequency of 2 gives a steady throughput of nearly 50%. In the case of very high PER of 10^{-1} , it is seen that even RF of 4 is not sufficient to enhance the throughput and with RF of 8 only steady throughput of 47% is obtained.

From the simulation experiments, it can be concluded that the Adaptive Redundant Retransmission Technique is very efficient under high error conditions and provides an appreciable throughput. Moreover, it does not have any implication when the PER is less.

6. Analysis of the Timeout Algorithm

It has been discussed that the timeout algorithm used in DR_TCP is different from conventional timeout algorithm. In this section, we analyze the performance of the timeout algorithm. For this analysis we consider the LUNAR Exploration simulations with RTT = 2.560 sec and check the frequency of timeout events for different packet error rates.

From the results, it can be seen that, when the packer error rate is less no timeout events were recorded. This shows the efficiency of the DR-TCP timeout algorithm and confirms that it does not lead to an underestimation of the waiting time and very effectively estimates the

timeout values. It can also be seen that as the packet error rates become very high, the timeout events start occurring with a very high timeout frequency for the 10^{-1} case, as seen in the first column of the **Table 12** where no redundant retransmission is done. This also explains the degradation in throughput seen in the 10^{-1} case for DR-TCP with utilization dropping to 32% when timeout frequency reaches to 250.

When the Adaptive Redundant Retransmission technique is applied with RF = 2 the Timeout frequency drops to 102 and to 24 with RF = 4. This has been reflected in the enhancement of the performance of the protocol with increasing RF. With very high RF values timeouts increase because of high redundant packets. This shows the inappropriateness of very high RF like 16 and substantiates the rationale of the redundant retransmission technique.

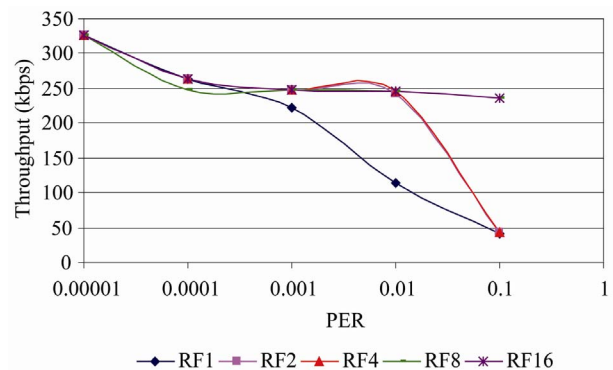
**Figure 5. Throughput of DR-TCP for RTT = 26.21 mins.**

Table 11. Percentage bandwidth utilization of DR-TCP with retransmission frequency for RTT = 26.21 mins.

PER	RF1	RF2	RF4	RF8	RF16
10^{-5}	65.105	65.1044	65.1028	65.1	65.094
10^{-4}	52.838	52.6822	52.8236	49.7136	52.7666
10^{-3}	44.3368	49.569	49.4858	49.5848	49.5848
10^{-2}	22.6358	48.2128	49.0482	49.0486	49.0486
10^{-1}	8.1972	8.5392	8.7796	47.0486	47.0048

Table 12. Timeout event frequency for lunar network.

PER	RF1	RF2	RF4	RF8	RF16
10^{-6}	0	0	0	0	0
10^{-5}	0	0	0	0	0
10^{-4}	0	0	0	0	0
10^{-3}	0	0	0	0	0
10^{-2}	12	5	5	14	41
10^{-1}	250	102	24	31	27

6.1. Adaptive Redundant Retransmission Technique

From the simulation experiments, it has been established that with increasing values of retransmission frequency, the throughput increases till a point after which the throughput starts to decrease. DR-TCP binomially increases the retransmission frequency and checks if that leads to an increase in throughput. Whenever the Packet Error Rate is detected to be higher than 10^{-3} , redundancy is added in the retransmitted packets and the throughput so obtained can be checked. The point where a decrease in throughput is achieved is considered the optimum point. For example, in the Lunar Exploration case a increase from RF 4 to 8 decreases the throughput, so 4 can be considered as the desired RF value for that point of time. Similarly, in the case of Mars Exploration Network an increase of RF from 8 to 16 marginally reduces the throughput so 8 can be considered as the desired retransmission frequency. After a base value is obtained, the retransmission frequency may be varied linearly within its range to get the optimal value.

7. Conclusions

In this paper, a Transport Protocol suitable for the Planetary Exploration Missions that need point-to-point communication is proposed. The contributing factors that are responsible for the performance degradation of conventional transport protocols under high delay cases have been critically analyzed. The results of the analysis have been used in designing a delay resistant Transport Pro-

ocol. Using the ns2 implementation the Delay Resistant TCP is extensively simulated and the RTT independent nature of the protocol clearly been exhibited. DR-TCP gives good performance under very high delays generally encountered in Lunar or Mars exploration missions. From the analysis of DR-TCP results, it has been observed that though the protocol is RTT independent its performance degrades drastically at high error conditions. To circumvent this degradation a novel Adaptive Redundant Retransmission Technique is proposed and it has been analyzed considering different values of retransmission frequency. With the application of the technique it has been shown that, the protocol can always provide nearly 50% bandwidth utilization even under high error conditions. Moreover, the application of this technique does not degrade the performance of the protocol when error is less. The RTT independent nature of the protocol along with the ability to provide sustained throughput at high error conditions makes it a good candidate to be used in links with very high RTT typically seen in planetary exploration networks. Minimum changes are necessary in the sender and receiver protocol stacks.

8. References

- [1] O. B. Akan, J. Fang and I. F. Akyildiz, "Performance of TCP Protocols in Deep Space Communication Networks," *IEEE Communications Letters*, Vol. 6, No. 11, November 2002, pp. 478-480. doi:10.1109/LCOMM.2002.805549
- [2] V. Cerf, S. Burleigh, *et al.*, "Interplanetary Internet (IPN): Architectural Definition," draft-irtf-ipnrg-arch-00.txt, May 2001.

- [3] V. Cerf, S. Burleigh, *et al.*, "Delay-Tolerant Network Architecture: The Evolving Interplanetary Internet," draft-irtf-ipnrg-arch-01.txt, August 2002.
- [4] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang and W. L. Su, "Interplanetary Internet: State-of-the-Art and Research Challenges," *Computer Networks*, Vol. 43, No. 2, October 2003, pp. 75-113.
- [5] I. Psaras, G. Papastergiou, V. Tsaousidis and N. Peccia, "DS-TP: Deep-Space Transport Protocol," *Proceedings of IEEE Aerospace Conference*, 20 May 2008, Big Sky, pp. 1-13.
- [6] T. de Cola, H. Ernst and M. Marchese, "Performance Analysis of CCSDS File Delivery Protocol and Erasure Coding Techniques in Deep Space Environments," *Computer Networks*, Vol. 51, No. 14, May 2007, pp. 4032-4049. doi:10.1016/j.comnet.2007.04.015
- [7] O. B. Akan, J. Fang and I. F. Akyildiz, "TP-Planet: A Reliable Transport Protocol for Interplanetary Internet," *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 2, February 2004, pp. 348-361. doi:10.1109/JSAC.2003.819985
- [8] V. Cerf, S. Burleigh, *et al.*, "RFC 4838, Delay-Tolerant Networking Architecture," IRTF DTN Research Group, April 2007.
- [9] K. Scott, S. Burleigh, *et al.*, "RFC 5050, Bundle Protocol Specification," IRTF DTN Research Group, November 2007.
- [10] L. Wood, J. McKim, *et al.*, "Saratoga: A Scalable File Transfer Protocol," draft-wood-tsvwg-saratoga-05, May 2010.
- [11] S. Burleigh, M. Ramadas, *et al.*, "RFC 5325, Licklider Transmission Protocol Motivation," IRTF DTN Research Group, September 2008.
- [12] M. Ramadas, S. Burleigh, *et al.*, "RFC 5326, Licklider Transmission Protocol Specification," IRTF DTN Research Group, September 2008.
- [13] S. Farrell, M. Ramadas, *et al.*, "RFC 5327, Licklider Transmission Protocol—Security Extensions," IRTF DTN Research Group, September 2008.
- [14] K. Bhasin, J. Hayden, *et al.*, "Advanced Communication and Networking Technologies for Mars Exploration," *Proceedings of 19th International Communications Satellite Systems Conference*, 17-20 April 2001, Toulouse, pp. 1-10.
- [15] D. Rossi, C. Testa, S. Valenti and L. Muscariello, "LEDBAT: The New BitTorrent Congestion Control Protocol," *Proceedings of the 19th International Conference on Computer Communications and Networks*, 2-5 August 2010, Zurich, pp. 1-6.
- [16] M. Sarkar, K. K. Shukla and K. S. Dasgupta, "A Proactive Transport Protocol for Performance Enhancement of Satellite based Networks," *International Journal of Computer Applications*, Vol. 1, No. 16, February 2010, pp. 100-107.
- [17] M. Sarkar, K. K. Shukla and K. S. Dasgupta, "Network State Classification Based on the Statistical Properties of RTT for an Adaptive Multi State Proactive Transport Protocol for Satellite Based Networks," *International Journal of Computer Networks & Communications*, Vol. 2, No. 6, November 2010, pp. 155-174.
- [18] R. Jain, D. Chiu and W. Hawed, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," DEC, Research Report TR-301, 1984.
- [19] L. Wood, C. Peoples, G. Parr, B. Scotney and A. Moore, "TCP's Protocol Radius: The Distance where Timers Prevent Communication," *Proceedings of 3rd International Workshop on Satellite and Space Communications*, 13-14 September 2007, Salzburg, pp. 163-167. doi:10.1109/IWSSC.2007.4409409
- [20] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP Selective Acknowledgment Options," RFC 218, April 1996.
- [21] UCB/LBNL/VINT Network Simulator. <http://www.isi.edu/nsnam/ns/>
- [22] L. S. Brakmo, S. O Malley, L. L. Peterson, *et al.*, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *ACM SIGCOMM Computer Communication Review*, Vol. 24, No. 4, October 1994, pp. 24-35. doi:10.1145/190809.190317
- [23] L. A. Grieco and S. Mascolo, "A Congestion Control Algorithm for the Deep Space Internet," *Space Communications*, Vol. 20, No. 3-4, 2006, pp. 155-160.
- [24] R. H. Wang, W.-T. Hsu, X. Wu, T. T. Wang and X. B. Wang, "Experimental and Comparative Analysis of Channel Delay Impact on Rate-Based and Window-Based Transmission Mechanisms over Space-Internet Links," *Proceedings of IEEE International Conference on Communications*, 19-23 May 2008, Beijing, pp. 2990-2994.
- [25] R. H. Wang, A. Ayyagari, X. Wu, B. Sun and W.-T. Hsu, "Experimental Performance Comparison of Rate-Based and Store-and-Forward Transmission Mechanisms over Error-Prone Cislunar Communication Links," *Proceedings of IEEE International Conference on Communications*, 24-28 June 2007, Glasgow, pp. 4518-4522.