DS-TP: Deep-Space Transport Protocol

Ioannis Psaras[‡], Giorgos Papastergiou[‡], Vassilis Tsaoussidis[‡] and Nestor Peccia[⋆]
[‡]Department of Electrical and Computer Engineering
Democritus University of Thrace
12 Vas. Sofias Str.,
67100, Xanthi, Greece
(ipsaras,gpapaste,vtsaousi)@ee.duth.gr
[⋆]Mission Operations Center
European Space Agency (ESA)
Darmstadt, Germany
nestor.peccia@esa.int

Abstract—We present Deep-Space Transport Protocol (DS-TP), a new reliable protocol for deep-space communication links. DS-TP's main advantage is its ability to complete file transfers faster than conventional TCP, SCPS-TP and Saratoga. Therefore, missions with small connectivity time are greatly favored.

Deep space communication links are characterized by long propagation delays, high BERs, intermittent connectivity (i.e., blackouts) and bandwidth asymmetries. Common approaches to deal with the above unique characteristics are: rate-based, open-loop protocols to deal with huge propagation delays; regular retransmissions to deal with high BERs; transmission suspension to deal with blackouts; SNACKs to deal with bandwidth asymmetries. We adopt some of the above approaches, namely, the open-loop, rate-based transmission and the SNACKs and focus on the optimization of the rest, namely, the retransmission strategy of the transport protocol to deal either with high BERs or with blackouts.

More precisely, DS-TP includes the Double Automatic Retransmission (DAR) technique. DAR sends each packet twice, importing some intentional delay (R_d) between the original transmission and the retransmission. Therefore, in the presence of communication gaps (i.e., errors or blackouts), corrupted packets will eventually be replaced by the same correct packets that arrive with delay R_d. R_d, however, is much smaller than the traditional TCP-RTO value.

Our theoretical performance evaluation results reveal that DS-TP presents high potential for deployability. In particular, we show that for PER=50%, DS-TP completes a file transfer in half time of a conventional protocol.

TABLE OF CONTENTS

1-4244-1488-1/08/$25.00 ©2008 IEEE
IEEEAC Paper #1160, Version 5 Updated 13/12/2007.

1. INTRODUCTION

The success of a space mission is heavily dependend on the performance of its communication means. Imagine, for example, space objects on the surface of Moon or Mars (e.g., rovers, robots etc.) gathering scientific data; this data needs to traverse a (deep) space link to reach a ground station antenna on Earth. Scientific data transfers take place when the two end-points of the communication path are within the line-of-sight. In this context, communication or connectivity time becomes a crucial point for the success of the mission. When communication time is in the order of some minutes per week, data has to be buffered for later transmission. Buffering requirements in that case, however, become extraordinary. Provided that space objects cannot be equipped with such a great amount of memory, scientific data is, finally, discarded (i.e., lost).

There are two possible optimization approaches in order to avoid extensive buffering requirements: i) leasing of communication time or, ii) increase the amount of data transferred within specific timelines. Communication time, however, cannot be increased easily since: i) there exists a mission-specific, hard upper limit for the available communication time, ii) resource conflicts with other missions may reduce further the above-mentioned limit and iii) buying more time (from other space agencies) is expensive. Moreover, communication times for current space missions are reserved by human-operated management systems before the mission's execution. Among others, this means that routing of scientific data is static and pre-determined.

Lately, there is a common consensus among space agencies that the performance of space communication networks will increase if a decentralized, dynamic (IP) routing model is adopted [1]. Indeed, such a scheme will do increase connectivity time, but only in the near Earth (e.g., Earth to Moon) space environment, where link propagation delays are in the order of a few seconds. As the distance to the space object increases, the propagation delay of the link increases as well, rendering the use of IP prohibitive. In other words, *IP cannot operate over deep-space links*. Thus, regarding deep-space communications (e.g., Earth to Mars), the only alternative optimization approach is to increase the amount of data transferred within specific timeframes[1] (i.e., increase throughput performance).

In the current study, we focus on InterPlanetary communication links, their characteristics and the performance of transport protocols over such transmission links. We identify the differences between Internet and deep-space links and try to deal with them, accordingly. Our solution framework is incorporated within the Deep-Space Transport Protocol (DS-TP).

Deep space transmission links, in contrast to conventional Internet links, are characterized by:

- **Very High Propagation Delays.** The propagation delay between Earth and Mars varies between 8 and 40 minutes depending on the orbital location of the planets [2], [3]. Propagation delays to outer space planets increases with the distance and may become higher than 100 minutes.
- **High Link Bit Error Rates.** Deep space links are characterized by extremely high bit error rates, which may be up to 10^{-3} [2], [3].
- **Intermittent Connectivity.** Planetary bodies, asteroids or spacecraft may periodically interrupt the communication link between the path endpoints [3], [4].
- **Bandwidth Asymmetry.** Asymmetry between the forward and the return path bandwidth may be up to 1000:1 [3], [5].

It is widely accepted that regular TCP cannot operate efficiently (if at all) under such conditions. Akan et al. have shown in [5], that TCP needs 120 minutes to reach Slow Start Threshold equal to 20 packets, over a 20 minute Round Trip Time (RTT) path. Obviously, longer RTT paths degrade TCP's performance further. Moreover, since TCP was designed to operate over wired transmission channels, where the link error rate is insignificant, the protocol cannot cope with increased link errors and blackouts [6], [7]. Finally, TCP's transmission rate depends largely on the receiver's feedback, which is received at the sender side in the form of acknowledgments (ACKs). TCP sends one ACK for each successfully received data packet. On the presence of bandwidth asymmetry, however, the large number of ACKs that need to be sent back to the sender, will cause congestion on the reverse path, reducing TCP's transmission rate.

Clearly, we have to move towards a new protocol-design era, in order to achieve high link utilization for deep space communications. For example, the transport protocol's transmission rate needs to be decoupled from either positive or negative feedback from the receiver (i.e., feedback is already too "old" or in other words, "news" is already late). Moreover, there is no need for congestion control, since deep space data transfers are pre-scheduled. Therefore, the bandwidth fair-share is known a priori.

Common approaches to deal with the above unique characteristics are: rate-based, open-loop protocols to deal with huge propagation delays; regular retransmissions to deal with high BERs; transmission suspension to deal with blackouts; SNACKs (i.e., Selective Negative ACKs) to deal with bandwidth asymmetries. We adopt some of the above approaches, namely, the open-loop, rate-based transmission and the SNACKs and focus on the optimization of the rest, namely, the retransmission strategy of the transport protocol to deal either with high BERs or with blackouts.

More precisely, DS-TP implements the Double Automatic Retransmission (DAR) technique. DAR sends each packet twice, importing some delay (R_d) between the original transmission and the retransmission. Therefore, in the presence of communication gaps (i.e., errors or blackouts), corrupted packets will eventually be replaced by the same correct packets that arrive with delay R_d. R_d, however, is much smaller than the traditional TCP-RTO value. We show that DS-TP can complete file transfers much faster than conventional transport protocols. In particular, as the channel PER increases, DS-TP's response time (for retransmission) decreases, leading to faster file delivery time.

The rest of the paper is organized as follows: in Section 2, we briefly discuss alternative proposals for deep-space data transfers. In Section 3, we present the main operational functionalities of the Deep-Space Transport Protocol. In Section 4, we analyze theoretically the performance of DS-TP's main components. Section 5 includes our Protocol Evaluation Framework, while in Section 6, we compare the performance of DS-TP against a conventional fixed-rate transport protocol. Finally, in Section 7, we conclude the current study.

2. Related Work

Although deep-space communications is a relatively new research topic, there exist already a number of proposals regarding transport layer networking over deep-space links. In this section, we briefly review these proposals.

The most well-known famous transport protocol, which is designed for reliable data transmission over deep space links is TP-Planet [8]. In contrast to DS-TP, the main functionality of TP-Planet is a probing congestion detection and control mechanism to deal with congestion losses. Moreover, TP-Planet uses a Blackout State procedure to deal with blackouts and the delayed SACK strategy to deal with bandwidth

[1] A possible alternative to that, is the extension of ground or space infrastructure in order to extend connectivity and paths. However, since this approach requires significant investments, it is not considered, presently.

asymmetry. More precisely, TP-Planet uses a rate-based Additive Increase Multiplicative Decrease (AIMD) congestion control, whose operation depends on the decision of the congestion detection mechanism. Deep-space communications, however, at least presently, operate with static, pre-scheduled management procedures, which are fixed long before the mission itself. Therefore, congestion control is not really needed, since flow multiplexing over deep-space links does not exist. In that context, TP-Planet seems to be *"over-qualified"* for deep-space data transfers.

A similar approach, which comes from the same authors, is the unreliable RCP-Planet [9] protocol. RCP-Planet incorporates a probing rate control scheme to cope with link congestion and error rate, in conjunction with a packet-level FEC. RCP-Planet also deploys a Blackout state procedure and FEC block-level ACKs to address bandwidth asymmetry. RCP-Planet's main target is the delivery of real-time application data either to the ground or to the satellite, spacecraft etc. The term real-time, however, does not really exist for channels with propagation delays in the order of tens or hundreds of minutes. Although both of the above protocols have different design goals than DS-TP, we include them here, since they are composed by mechanisms, such as the Blackout state, which present high potential for deployability in other protocols as well.

Space Communications Protocol Standards - Transport Protocol (SCPS-TP) [10] is a protocol developed by the Consultative Committee for Space Data Systems (CCSDS) [11] for space communications. SCPS-TP is based on the widely used Transmission Control Protocol (TCP) and includes a set of modifications and extensions to deal with the unique constraints of deep space communication links. SCPS-TP operates in one of the following two modes: i) the Van Jacobson Congestion Control mode, which incorporates the TCP-Vegas [12] approach and ii) the Open Loop Rate Control mode. The Open Loop Rate Control mode is based on the "corruption-experienced" signal from the receiver side and assumes that there is no congestion on the link. Additionally, to deal with bandwidth asymmetry SCPS-TP uses Selective Negative Acknowledgments, which in contrast to simple Negative ACKs (NAKs) are able to identify multiple holes in the receiver's sequence number space.

Saratoga [13] is a reliable rate-based UDP/IP file transfer protocol, capable of transferring efficiently both small and very large files. It has been developed by Surrey Satellite Technology Ltd (SSTL) and it is used for mission imaging data. Saratoga was designed for dedicated point-to-point links between peers; it focuses on transferring data efficiently to the next hop, when link connectivity is available. Saratoga achieves efficient transmission by sending out data packets at the line rate. It also uses a negative acknowledgment strategy in order to deal with channel bandwidth asymmetries. Saratoga can be used as a convergence layer to exchange Delay Tolerant Networking bundles [14], [15] between peer nodes [13].

A similar file-oriented protocol is the CCSDS File Delivery Protocol (CFDP) [16], which is mainly an application layer protocol, including transport layer functionalities as well. File transmission can be executed reliably (acknowledged mode) or unreliably (unacknowledged mode). CFDP provides file delivery services i) across a single link (referred to as Core Functionalities) and ii) over more complex topologies, where CFDP provides subsequent transmissions of files between intermediate nodes, which end up to the destination node (i.e., Extended Procedures/Store-and-Forward Overlay). CFDP includes four modes for sending Negative Acknowledgments (i.e., Deferred, Immediate, Prompted and Asynchronous) and uses positive Acknowledgments (ACKs) as well, to ensure the receipt of critical PDUs.

Similarly to Saratoga, the Licklider Transmission Protocol (LTP) [17] is a point-to-point protocol applied as a DTN convergence layer. LTP can transfer unnamed blocks of data and introduces the concept of partial reliability by dividing each block of data into two parts: the reliable *"red"* part and the unreliable *"green part"*. Moreover, *laconic* acknowledgments are sent only upon encountering explicit solicitations for reception reports (checkpoints) in the sequence of incoming data segments of the red part of the block. Deferred Transmission is possible as well, in case the communication link is not available.

3. DS-TP: DEEP-SPACE TRANSPORT PROTOCOL

In this section, we initially discuss the main operational properties of the Deep-Space Transport Protocol, as well as the rationale associated with our choices. Next, we describe in detail the functionality of the *Double Automatic Retransmission* technique and finally, we give implementation details and parameter settings regarding DS-TP's SNACK strategy.

Basic Components

DS-TP has the following fundamental characteristics:

1. **Rate-based transmission.** Data routing and forwarding in current space missions, take place on a static, pre-scheduled, human-operated manner. Dynamic (IP) routing is a strong alternative to the current approach, which is expected to increase connectivity time and reduce management and scheduling costs. This approach, however, does not apply for deep-space missions, where the propagation delay of the end-to-end path becomes extremely high; routing table updates, for example, will not provide trustworthy information. Therefore, deep-space missions will continue to operate based on a static, predetermined manner, where the bandwidth of the transmission link is known a priori. That said, a fixed-rate transmission tactic allows for high link utilization, without forcing transmission rate increase, which finally leads to congestion losses.

Moreover, TCP's closed loop, ACK-clocked transmission tactic proves to be inefficient for propagation delays in the order of minutes. That is, feedback arrives at the sender side several minutes later, at a time when the information included in the feedback packet(s) (i.e., ACK(s)) may already be useless (e.g., triple duplicate ACKs will trigger congestion avoidance and recovery mechanisms, but the actual congestion event has happened several minutes earlier).

2. **Mixed ACK - SNACK Strategy.** Clearly, the above situation call for decoupling of the ACK role from transmission rate adjustments. The positive ACK is used by DS-TP for retransmission buffer space release at the sender side. Moreover, provided that deep-space links are characterized by bandwidth asymmetries, sending one ACK for each incoming packet may cause congestion on the reverse path and consequently transmission rate reduction. A common approach to deal with bandwidth asymmetries in satellite and space communications is the use of Selective Negative ACKs (SNACKs). In contrast to simple Negative ACKs (NAKs), SNACKs are able to signal for multiple holes at the receiver's buffer.

In conclusion, DS-TP sends positive ACKs to trigger buffer space release at the sender side, whenever there are no holes at the receiver's sequence number buffer space and SNACKs to either allow for network measurements or trigger retransmission of lost segments. The detailed operation and functionality of SNACKs is presented later in this section. Although one may argue that the mixed ACK-SNACK strategy, used here, does not reduce the number of (SN)ACK packets inserted at the slow reverse link, we note that loss of a (SN)ACK, due to congestion for example, neither causes reduction of the sender's transmission rate, nor prevents buffer space release at the sender's network interface.

3. **Retransmission Policy.** DS-TP implements a novel retransmission technique, called *Double Automatic Retransmission* (DAR), which allows for fast and efficient "hole-filling" at the receiver's buffer. DAR sends each packet twice, importing some delay (R_d) between the original transmission and the retransmission. Therefore, in the presence of link errors, corrupted packets will eventually be replaced by the same correct packets that arrive with delay R_d. R_d, however, is much smaller than the traditional TCP-RTO value. The probability that both the original and the retransmitted packets are lost is x^2, where x is the link PER and $x < 1$. For example, if one out of three packets is lost, DS-TP's transmission sequence is 1-2-**1**-3-4-**2**-5-6-**3** etc. R_d is initially set to a small value that corresponds to high PER (e.g., 50%) and is adjusted according to the actual PER, based on network measurements. DAR is presented in detail below.

Double Automatic Retransmission (DAR)

Transmission Sequence—As we have already mentioned earlier, DS-TP injects data packets into the transmission link in a pre-determined, fixed rate (i.e., the *Actual Rate*). Apart from the *Actual Rate*, the DS-TP sender keeps one extra variable, called *Retransmission Rate*, which regulates the retransmission rate of the protocol. The *Retransmission Rate* is set according to the link error rate, the measurement of which is discussed in the following sections. The DS-TP sender keeps, apart from the regular *current sequence number* (`c_seqno`) variable, the *retransmission sequence number* (`r_seqno`), as well. Similarly, to the current sequence number, which indicates the maximum packet number that has been *sent* so far, the retransmission sequence number holds the maximum packet number that has been *retransmitted* from DAR, so far.

DS-TP transmits each packet twice importing some delay R_d between the original transmission and the retransmission. The delay between the original transmission and the retransmission, R_d, is implemented in DS-TP in terms of packets and depends on the channel packet error rate. For example, if $error\_rate = 20\%$, which means that one out of five packets is corrupted due to link errors, DS-TP transmits one redundant packet every four original packets (see Figure 1). In other words, one redundant packet is transmitted every $\frac{1}{error\_rate} - 1$ original packets. The retransmission sequence number is, thus, given by:

$$r\_seqno = \frac{c\_seqno - 1}{\frac{1}{error\_rate} - 1}. \quad (1)$$

Obviously, whenever Equation 1 leads to a non-integer value, `r_seqno` is rounded downwards to the closest integer value.

Therefore, a packet with sequence number `c_seqno` will be retransmitted after `diff_pkts` number of packets, according to the following formula:

$$diff\_pkts = [(\frac{1}{error\_rate} - 1) \cdot c\_seqno] - r\_seqno. \quad (2)$$

At the receiver side this is interpreted as follows: once a hole at the receiver's buffer is detected, which corresponds to packet with sequence number `c_seqno`, the receiver expects this packet to arrive after $diff\_pkts + 1$ number of packets.

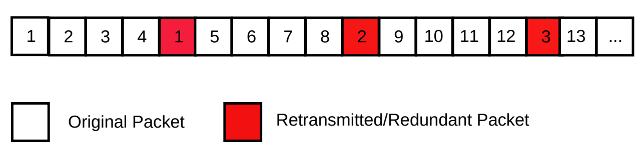


Figure 1. Example Packet Transmission Sequence

Summarizing, if $error\_rate = 20\%$, the packet with `c_seqno` = 3 will be retransmitted after 12 packets, according to Equation 2, since at that time `r_seqno` = 0, according to Equation 1. The packet transmission and retransmission sequence, in that case, is shown if Figure 1.

In other words, DAR transmits redundant packets with $error\_rate$ Mbps. The original-packet transmission rate is, thus, reduced to $Original\ Packet\ Rate = Link\ Rate - error\_rate$. Reffering to the previous example, we have that $Original\ Packet\ Rate = 80\% \cdot Link\ Rate$.

SNACK types and their Functionality—An important component of DS-TP is its *Link Error Rate Measurement* functionality. For that purpose, DS-TP exploits the receiver's feedback, which arrives at the sender side in the form of mixed ACKs and SNACKs. As we have already mentioned before, positive ACKs are used for releasing space at the sender's retransmission buffer. Moreover, DS-TP uses two types of Selective Negative ACKs, namely $SNACK_1$ and $SNACK_2$, whose main functionality is discussed below.

- $SNACK_1$: The DS-TP receiver produces $SNACK_1$, whenever it receives a new data packet *and* at the same time, one or more holes exist in its receiving buffer space. Upon arrival of $SNACK_1$ at the sender side, the sender *does not* retransmit any of the missing packets, indicated by the $SNACK_1$. Instead, the DS-TP sender uses the information included in $SNACK_1$ to calculate the link error rate. In particular, each $SNACK_1$ includes a cumulative positive ACK, which acknowledges arrival of some packets at the receiver side. The ratio of the number of holes (and their size), included in $SNACK_1$, over the total number of packets ACKed until that time, constitutes a close approximation of the link error rate experienced by the receiver, until that time. The rationale behind this behavior (i.e., no retransmission attempt upon $SNACK_1$ arrival at the sender side) is that DAR will automatically retransmit the missing packets, according to Equation 1. This retransmission, however, will take place earlier than the $SNACK_1$ arrival at the sender side. Therefore, in case the missing packet is not corrupted for a second time, then the redundant packet will arrive faster than the hypothetical retransmission triggered by $SNACK_1$[2]. The probability that the redundant packet will be corrupted again is reduced to x^2, where x is the link error rate and $x < 1$.
- $SNACK_2$: Being aware of the sender's automatic retransmission policy (i.e., Equation 1), the DS-TP receiver expects arrival of the redundant packet, according to Equation 1. In case, the redundant packet does not arrive, whithin that interval, a $SNACK_2$ is sent. In contrast to $SNACK_1$, $SNACK_2$ triggers immediate retransmission of the missing segment(s).

There are two alternative approaches in order to deal with corrupted packets, whose retransmission is triggered by $SNACK_2$:

1. The DS-TP receiver sends a second $SNACK_2$ in order to trigger retransmission of the lost packet. In that case, the receiver has to schedule a timer for each incoming packet, whose retransmission was triggered by $SNACK_2$.

[2]DAR achieves faster retransmission than $SNACK_1$ under specific conditions discussed in the next section (i.e., Section 4).

2. The DS-TP sender keeps one separate timer for each retransmission triggered by $SNACK_2$.

In both cases, the retransmission timeout value (either for $SNACK_2$ or for the actual packet) is set approximately equal to the path RTT. Further investigation is needed in order to choose the most appropriate response function.

Finally, note that the cumulative nature of ACKs and SNACKs reduces the impact of ACK or SNACK loss, due to congestion or corruption on the reverse path. Therefore, we do not apply DAR on the reverse path (i.e., for (SN)ACKs), since DS-TP's performance is not affected by (SN)ACK loss.

In Table 1, we include the main acronyms used throughout the rest of the paper.

Table 1. `c_seqno` Boundary between DAR and $SNACK_1$ Retransmission (variable bandwidth)

Symbol	Meaning
c_seqno	Current Sequence Number
r_seqno	Retransmission Sequence Number
x	Link Tranmission Rate
fs	File Size
$error\_rate$	Link Error Rate
y	$error\_rate$

4. DS-TP Theoretical Evaluation

We attempt to evaluate, theoretically, the performance of DS-TP. In order to derive some initial evaluation results regarding the performance of the proposed transport protocol, we assume that the link error rate remains constant during the data transfer.

We depart from Equation 1, in order to monitor the relation between the current sequence number and the retransmission sequence number, which constitutes a fundamental issue regarding the performance of DS-TP. The results are presented in Figure 2. We observe that the relation between `c_seqno` and `r_seqno` depends on the link error rate. In other words, the advantage of faster retransmission attempt, due to DAR, increases with the error rate.

Graphical representation of Equation 2 is presented in Figure 3. We see that the difference between the original transmission and the retransmission increases linearly with the error rate and may reach extraordinary high levels, as the (current) sequence number increases. For example, we see in Figure 3 that packet 1000 will be retransmitted after 9000 packets, when the link error rate is 10%. Obviously, this difference increases further for higher sequence numbers. As the link error rate increases, we observe that the difference between the original transmission and the retransmission decreases (e.g., when link error rate is 50%, the difference between the original transmission and the retransmission is constant and equal to 1 packet). This fact further strengthens our previous claim

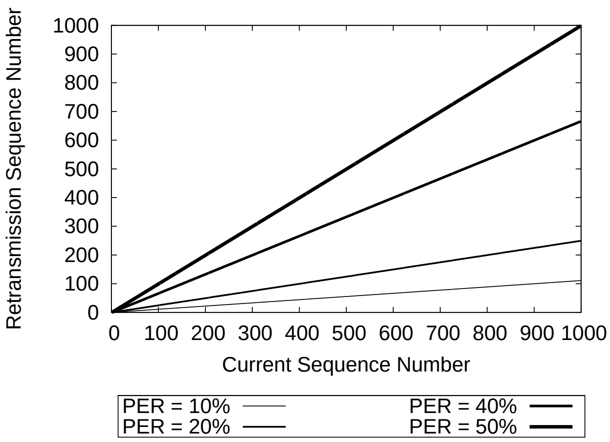


Figure 2. Current vs. Retransmission Sequence Number

that DAR provides faster retransmission services as the link error rate increases.

One straightforward issue that arises from the above discussion is *"How long will it take DAR to retransmit each specific packet?"*. Or in other words *"Is it worth to wait for DAR retransmission instead of retransmitting upon SNACK$_1$ arrival at the sender side?"*. In the following, we attempt to answer the above questions.

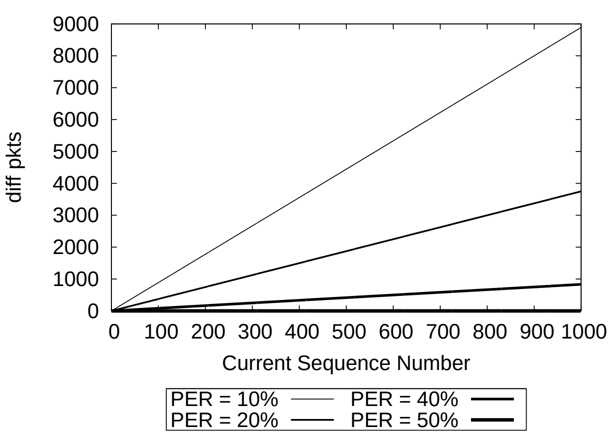


Figure 3. Number of Packets between Original Transmission and Retransmission

The time requirement for transmission of a specific number of packets over a transmission link depends mainly on the transmission speed of the link. For example, a x Mbps link can transfer $\frac{x}{8}$MBps, or equivalently $\frac{1024 \cdot x}{8}$KB/s (i.e., packets/second). Therefore, `diff_pkts` require $\frac{8 \cdot diff\_pkts}{1024 \cdot x}$ seconds to be transmitted over a xMbps link:

$$diff\_time = \frac{8 \cdot diff\_pkts}{1024 \cdot x}. \tag{3}$$

In Figure 4, we depict the evolution of the time interval between the original transmission and the retransmission for specific number of `diff_pkts`, according to Equation 3 for various transmission link speeds. We see that as the number of packets between the original transmission and the retransmission increases (e.g., 10,000 packets or more), the time interval between the two packets increases as well and may reach the order of 400 seconds or more. In order to elaborate more on this issue and identify specific packets (i.e., sequence numbers), which experience extremely high retransmission periods, we replace `diff_pkts` in Equation 3 using Equations 1 and 2:

$$diff\_time = \frac{8 \cdot [(\frac{1}{error\_rate} - 1) \cdot c\_seqno - r\_seqno]}{1024 \cdot x} = \\ = \frac{8 \cdot [(\frac{1}{error\_rate} - 1) \cdot c\_seqno - \frac{c\_seqno - 1}{\frac{1}{error\_rate} - 1}]}{1024 \cdot x} \tag{4}$$

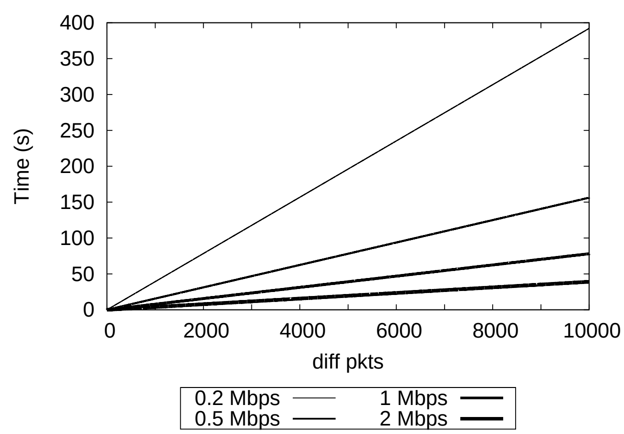


Figure 4. Time Interval between Original Transmission and Retransmission

We graph Equation 4 for various link speeds in Figure 5 and for various link error rates in Figure 6. In Figure 5, we see that the retransmission interval for a low speed link, which is quite common for deep-space communications, may be up to 1500 seconds (i.e., 25 minutes) for the 10,000th packet. When the reverse link propagation delay is smaller than the retransmission interval, depicted in Figure 5 or 6, then the DAR's functionality is cancelled and retransmission triggered by SNACK$_1$ should be adopted instead. In other words, if the forward link propagation delay is $Pr.\ D.\ For.$ and the reverse link propagation delay is $Pr.\ D.\ Rev.$, then Equation 5 should hold:

$$diff\_time \leq Pr.\ D.\ Rev. \tag{5}$$

Otherwise, it is more efficient to add retransmission functionality to SNACK$_1$, than wait for DAR to retransmit the corrupted packet.

In order to avoid delayed retransmission due to DAR, the DS-TP sender calculates `diff_time` according to the current sequence number (i.e., Equation 4) and schedules the retransmission attempts accordingly. In particular, if $diff\_time \leq$

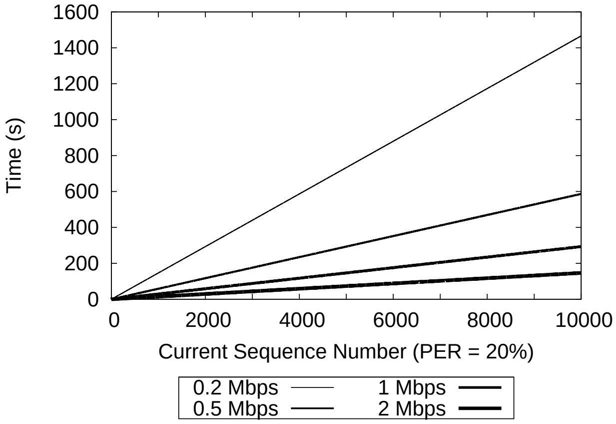


Figure 5. Time Interval between Original Transmission and Retransmission for Different Link Speeds (PER=20%)

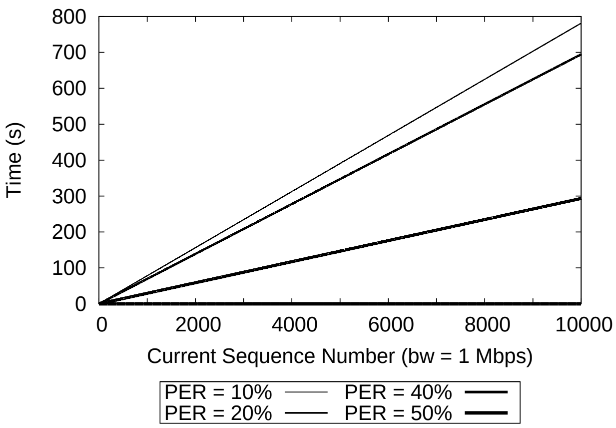


Figure 6. Time Interval between Original Transmission and Retransmission for Different Link Error Rates (bw=1Mbps)

$\frac{RTT}{2}$, then retransmissions take place following DAR. Otherwise, if $diff\_time > \frac{RTT}{2}$, then arrival of SNACK_1 at the sender side triggers immediate retransmission of lost/corrupted packets. We use Equations 4 and 5 in order to identify the sequence number (i.e., *c_seqno*) boundary that cancels DAR and triggers immediate retransmission upon SNACK_1 arrival at the sender side. Therefore, for a x Mbps link, we use Equations 4 and 5, which lead to:

$$\frac{8 \cdot [(\frac{1}{error\_rate} - 1) \cdot c\_seqno - \frac{c\_seqno - 1}{\frac{1}{error\_rate} - 1}]}{1024 \cdot x} \leq \frac{RTT}{2}, \tag{6}$$

which after some algebra gives:

$$c\_seqno \leq \frac{64 \cdot x \cdot RTT \cdot (\frac{1}{error\_rate} - 1) - 1}{(\frac{1}{error\_rate} - 1)^2 - 1}, \tag{7}$$

which actually holds only when $(\frac{1}{error\_rate} - 1)^2 - 1 > 0$ or $error\_rate < 50\%$[3].

[3] According to DAR operational properties, error rates higher than 50% infer that the retransmission rate should become higher than the original transmission rate. In the current version of DS-TP, we do not adopt such setting; we leave the evaluation of this approach as a subject of future work.

Therefore, for sequence numbers greater than the right hand-side of Equation 7, the DS-TP sender triggers retransmission of lost/corrupted segments upon SNACK_1 arrival at the sender side. At that point, the DS-TP sender replaces `r_seqno` with `c_seqno`:

$$r\_seqno \leftarrow c\_seqno. \tag{8}$$

From that point onwards, the DS-TP sender continues the regular Double Automatic Retransmission, but the retransmission sequence number resumes from the current sequence number, according to Equation 8.

In Tables 2 and 3, we present the switching point of the retransmission sequence number (i.e., Equation 8) for various link speeds, PERs and RTTs. We see that the difference in terms of sequence number progress (i.e., `diff_seqno`) between the original transmission and the retransmission up to the point when the DAR retransmission interval becomes higher than $\frac{RTT}{2}$, depends on the Packet Error Rate. In particular, *the higher the PER, the faster the DAR retransmission takes place*.

In terms of time, the `diff_time` (i.e., Equation 3) is, obviously, equal to $\frac{RTT}{2}$ (see Tables 2 and 3). The field `total_time` indicates the total time elapsed (at the sender side) from the beginning of the transfer up to the point when Equation 8 applies (i.e., $c\_seqno + r\_seqno$ number of packets have been transferred). `total_time` represents the transmission delay of `c_seqno` number of packets over the xMbps link, according to:

$$total\_time = \frac{8 \cdot (c\_seqno + r\_seqno)}{1024 \cdot x}. \tag{9}$$

Since the `total_time` depends on the `c_seqno`, it follows that `total_time` depends both on the RTT and on the PER, as well, according to Equation 7.

The `diff_seqno` field in the above Tables, reveals that the percentage of the retransmissions with respect to the original transmissions depends on the link error rate, according to Equation 1. Therefore, for a fixed amount of data (i.e., fixed file size), the retransmission overhead will always follow the above rule (i.e., Equation 1), in contradiction to alternative approaches (e.g., FEC [18], LTP [17]), which produce fixed amounts of overhead regardless of the link error rate. For example, a 15MB file transfer over a 0.2Mbps, 5 minutes round trip propagation delay and PER = 20% link will produce $\frac{1}{4} \cdot 15 = 3.75$MB of overhead, according to Equation 1.

There is one more salient point regarding the performance of DS-TP over deep-space links. That is, there exists a fragile association between the `total_time` required to trans-

Table 2. c_seqno Boundary between DAR and SNACK$_1$ Retransmission (variable bandwidth)

m ← mins PER = 20%	RTT = 5 m	RTT = 10 m	RTT = 50 m	RTT = 100 m
$x = 0.2Mbps$				
c_seqno	1,024	2,048	10,240	20,480
r_seqno	256	512	2,560	5,120
diff_seqno	25%	25%	25%	25%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0.83 m	1.66 m	8.33 m	16.66 m
$x = 0.5Mbps$				
c_seqno	2,560	5,120	25,600	51,200
r_seqno	640	1,280	6,400	12,800
diff_seqno	25%	25%	25%	25%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0.83 m	1.66 m	8.33 m	16.66 m
$x = 1Mbps$				
c_seqno	5,120	10,240	51,200	102,400
r_seqno	1,280	2,560	12,800	25,600
diff_seqno	25%	25%	25%	25%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0.83 m	1.66 m	8.33 m	16.66 m
$x = 2Mbps$				
c_seqno	10,240	20,480	102,400	204,800
r_seqno	2,560	5,120	25,600	51,200
diff_seqno	25%	25%	25%	25%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0.83 m	1.66 m	8.33 m	16.66 m

Table 3. c_seqno Boundary between DAR and SNACK$_1$ Retransmission (variable PER)

m ← mins $x = 1Mbps$	RTT = 5 m	RTT = 10 m	RTT = 50 m	RTT = 100 m
PER = 10%				
c_seqno	2,160	4,320	21,600	43,200
r_seqno	240	480	2400	4,800
diff_seqno	11.1%	11.1%	11.1%	11.1%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0,31 m	0,63 m	3,13 m	6,25 m
PER = 20%				
c_seqno	5,120	10,240	51,200	102,400
r_seqno	1,280	2,506	12,800	25,600
diff_seqno	25%	25%	25%	25%
diff_time	2.5 m	5 m	25 m	50 m
total_time	0.83 m	1.66 m	8.33 m	16,66 m
PER = 40%				
c_seqno	23,039	46,079	230,399	460,799
r_seqno	15,359	30,719	153,599	307,199
diff_seqno	66.6%	66.6%	66.6%	66.6%
diff_time	2.5 m	5 m	25 m	50 m
total_time	5 m	10 m	50 m	100 m
PER = 49%				
c_seqno	239,892	479,796	$2.39{\cdot}10^6$	$4.79{\cdot}10^6$
r_seqno	230,484	460,980	$2.30{\cdot}10^6$	$4.60{\cdot}10^6$
diff_seqno	96.1%	96.1%	96.1%	96.1%
diff_time	2.5 m	5 m	25 m	50 m
total_time	61.25 m	122.5 m	612.5 m	1225 m

fer a file and the link characteristics (i.e., RTT, link rate and link error rate). More precisely, if $total\_time < RTT$, then the whole file will have already been transferred (i.e., have left the sender), before the DS-TP sender has any information at all regarding missing packets at the receiver side. In that case, the link will remain empty for as long as it takes for the receiver's feedback (i.e., SNACK) to reach the sender side. In particular, the link will be underutilized for $RTT - total\_time$ units of time. Thus, in order for DS-TP to achieve high performance, the following Equation should hold:

$$total\_time \geq RTT \Rightarrow \frac{8 \cdot (fs + retr)}{1024 \cdot x} \geq RTT, \qquad (10)$$

where $fs, retr$ and x stand for "file size", "retransmissions" and "link rate", respectively. In order to depict the retransmission overhead with respect to the file size, we use and modify Equation 1 as follows:

$$retr = \frac{fs - 1}{\frac{1}{error\_rate} - 1} \simeq \frac{fs}{\frac{1}{error\_rate} - 1} = \frac{fs \cdot error\_rate}{1 - error\_rate} \qquad (11)$$

From Equations 10 and 11 we get:

$\frac{8 \cdot \frac{fs}{1 - error\_rate}}{1024 \cdot x} \geq RTT$, or

$$fs \geq 128 \cdot RTT \cdot x \cdot (1 - error\_rate) \qquad (12)$$

According to the above, in case the file size is smaller than the right handside of Equation 12, then the DS-TP sender will be notified about missing packets, only after the whole file has already been inserted into the transmission link. In general, according to DS-TP, the sender will receive feedback for the packet with sequence number x_seqno after $T(x\_seqno)$ time units, where:

$$T(x\_seqno) = \frac{x\_seqno - (c\_seqno + r\_seqno)}{\frac{1024 \cdot x}{8}} + RTT. \qquad (13)$$

In other words, after $T(x\_seqno)$ time units, the DS-TP sender will be transmitting the packet with sequence number n_seqno, according to:

$$n\_seqno = x\_seqno + \frac{1024 \cdot x}{8} \cdot RTT. \qquad (14)$$

There are several alternative approaches to follow in order to avoid channel underutilization. For example, once the file has left the sender side, while no feedback has arrived yet, the DS-TP sender can begin transmission of other files, if there are any available. Otherwise, the sender can begin transmission of the same file for a second time, in order to (potentially) fill some holes at the receiver's buffer faster than waiting for the ACK/SNACK$_1$ feedback. We consider, however, that such settings are implementation-specific and therefore are outside the scope of the current study.

5. Protocol Evaluation Framework

Due to limited protocol implementations in simulation environments, we were not able to compare DS-TP with alternative proposals in a simulation environment. Although DS-TP is already implemented in the ns-2 [19] simulation environment, time constraints did not allow for additional protocol implementations (e.g., LTP [17], Saratoga [13]) in order to compare DS-TP's performance comparatively with alternative protocols. Here, we attempt to comparatively evaluate, in a theoretical framework, the performance of a modified, simpler version of DS-TP with a protocol, whose functionality is very close to that of CFDP [16] and Saratoga [13]. What we actually achieve is the evaluation of the gains obtained due to the Double Automatic Retransmission, which constitutes DS-TP's main functionality enhancement against similar proposals for deep space data transfers.

In particular, we consider the *Fixed-Rate Transport Protocol* (FR-TP), whose main functionality is summarized as follows: the FR-TP sender sends data on a fixed rate according to the pre-scheduled line rate, similarly to DS-TP. The FR-TP receiver, responds with SNACKs in order to signal for holes in the incoming transmission sequence. To simplify the analysis, we consider that SNACKs are sent back to the sender, only after the whole file has already arrived at the receiver side. This operation is similar to the deferred mode of CFDP [16].

We modify DS-TP in order to operate in a similar manner. That is, the DS-TP sender sends data according to the predetermined channel rate; DAR transmits redundant packets according to its operational rules (i.e., Sections 3 and 4), apart from its SNACK triggered retransmission policy. In particular, the DS-TP receiver sends SNACKs for missing packets after receipt of the whole file, similarly to FR-TP. Although such modification clearly degrades DS-TP's performance, since the sender is informed about missing packets later than usual, we use this setup in order to obtain comparative evaluation results between FR-TP and DS-TP.

We evaluate the performance of the aforementioned protocols over a simple one-hop topology. Such topology represents a deep-space link from one planet to another and may very well be used in conjunction with the DTN Bundle protocol [15]. Obviously, the primar metric of interest is the time required for the whole file to be delivered at the receiver side. The protocols' retransmission overhead is considered as well, in order to operate within acceptable energy consumption boundaries.

6. DS-TP vs FR-TP

According to our Evaluation Framework, we attempt to find the time required for a file to be reliably transferred from the sender to the receiver side. We define a *Round* to be the end-to-end transmission of a specific amount of data. A *Round* is initiated by the data transmission from the sender side and is terminated once SNACKs are sent back to the sender. That said, a file transfer consists of several *Rounds*, during the first of which the original file is transmitted, while during the rest of the *Rounds*, the sender retransmits packets lost in previous *Rounds*.

In the following, we sketch the performance of FR-TP and DS-TP respectively. For simplicity, we assume that the link error rate, denoted as y (i.e., $y \leftarrow error\_rate$), remains constant throughout the duration of the file transfer. The analysis presented below, however, can be easily extended to apply for variable link error rates, as well. In all cases, we consider that a file of size fs has to be transferred across the deep space link to the receiver.

FR-TP

The FR-TP sender will begin the transmission of the file at the channel rate. After completion of the first round the sender will have transmitted fs MBs. During the first round, $fs \cdot y$ MBs are lost and will need to be retransmitted during the second round. Similarly, $(fs \cdot y) \cdot y$ MBs are lost during the second round and need to be retransmitted during the third round. During the n^{th} round, the FR-TP sender will need to retransmit $fs \cdot y^n$ MBs. We assume that once the following Equation holds, then the file transfer is complete:

$$fs \cdot y^n < 1\, packet \tag{15}$$

Therefore, FR-TP needs n_{frtp} rounds in order to complete the file transfer:

$$n_{frtp} = \log_y(y^n) = \log_y(\frac{1}{fs}) = \frac{\log \frac{1}{fs}}{\log y} \tag{16}$$

Whenever the above Equation leads to a non-integer value for n, n is rounded upwards. In terms of time, the transmission delay for the first round (i.e., fs MBs) is $\frac{8 \cdot fs}{1024 \cdot x}$. Similarly, the transmission delay for the second round (i.e., $fs \cdot y$ MBs) is $\frac{8 \cdot fs \cdot y}{1024 \cdot x}$ and for the third round (i.e., $fs \cdot y^2$ MBs) is $\frac{8 \cdot fs \cdot y^2}{1024 \cdot x}$. In general, the transmission delay for the whole file transfer, after n rounds is given by:

$$FR - TP\, Tran.\, Delay = \frac{8 \cdot fs \cdot \sum_{k=1}^{n} y^{k-1}}{1024 \cdot x}. \tag{17}$$

DS-TP

According to its operational properties, DS-TP will transmit both original and redundant data at the line rate. During the first round, DS-TP will transmit in total $fs + r_1$ MBs, where r_1 is the number of retransmitted packets during this round and is given by Equation 11. The data packets transmitted during the first round, consist of $fs - r_1$ packets that were sent only once and r_1 packets that were sent twice, according to DAR. Since the channel packet error rate is y and applies for the total number of packets, we have that $fs - r_1$ packets are lost/corrupted with probability y, while the rest r_1 packets are lost/corrupted with probability y^2, where $r_1 = fs \cdot \frac{y}{1-y}$ (Equation 11). We assume that the number of packets lost during the first round (and need to be retransmitted during the second round) equals a_1, where:

$$a_1 = (fs - r_1) \cdot y + r_1 \cdot y^2. \quad (18)$$

Substituting r_1 into Equation 18, we get that:

$$a_1 = fs \cdot y \cdot (1 - y) \quad (19)$$

Similarly, during the second round, where a_1 MBs are transmitted, $a_1 - r_2$ packets are lost with probability y, while r_2 packets are lost with probability y^2, where $r_2 = a_1 \cdot \frac{y}{1-y}$. Again, assuming that a_2 number of packets are lost during the second round, we have:

$$a_2 = (a_1 - r_2) \cdot y + r_2 \cdot y^2. \quad (20)$$

We explicitly state that for the purpose of our theoretical evaluation, we are using DAR for the *retransmitted packets as well*. Although DS-TP's initial design does not include usage of DAR for the retransmitted packets, since (SN)ACKs are sent during the file transfer and not when the file transfer is complete, the current setup applies in case a "Deferred" or "Prompted" ACK strategy is adopted [16].

Substituting r_2 into Equation 18, we get that:

$$a_2 = fs \cdot y^2 \cdot (1 - y)^2 \quad (21)$$

DS-TP will complete the file transfer, when $a_z < 1$, where $z = n - 1$. Generalizing Equations 19 and 21, we assume that the file transfer is complete, once the following equation holds:

$$fs \cdot y^n \cdot (1 - y)^n < 1\, packet \quad (22)$$

Hence, DS-TP needs n_{dstp} rounds to transfer a fs MBs file:

$$n_{dstp} = \log_{[y\cdot(1-y)]}[y \cdot (1 - y)]^n = \log_{[y\cdot(1-y)]}(\tfrac{1}{fs}) \Rightarrow$$

$$n_{dstp} = \frac{\log \frac{1}{fs}}{\log(y \cdot (1 - y))} \quad (23)$$

Note that Equation 22 does not account for the packets sent during the initial (i.e., first) round. In order to include the packets sent during the first round, we modify Equation 22 as follows, and we call these packets *Original*:

$$Original = fs \cdot y^{n-1} \cdot (1 - y)^{n-1}. \quad (24)$$

The transmission delay for the above file transfer depends on the total number of packets sent. In particular, DS-TP sends as many data as FR-TP does (i.e., *Original*), plus the redundant data sent by DAR (i.e., *DAR Trans.*). The total number of redundant packets can be modelled as follows. During the first round and according to the above, DAR transmits $r_1 = fs \cdot \frac{y}{1-y}$ packets. During the second round DAR transmits $r_2 = a_1 \cdot \frac{y}{1-y}$ packets or $r_2 = fs \cdot y^2$ packets, according to Equation 19. During the third round DAR transmits $r_3 = a_2 \cdot \frac{y}{1-y}$ packets or $r_3 = fs \cdot y^3 \cdot (1 - y)$, according to Equation 21. Generalizing, we find that after n rounds, DAR will have transmitted r_n number of packets:

$$r_n = fs \cdot y^n \cdot (1 - y)^{n-2}. \quad (25)$$

Therefore, the total number of packets transmitted by DAR during the whole file transfer is:

$$DAR\, Trans. = \textstyle\sum_{k=1}^{n} r_k = \sum_{k=1}^{n} fs \cdot y^k \cdot (1 - y)^{k-2} \Rightarrow$$

$$DAR\, Trans. = fs \cdot \sum_{k=1}^{n} y^k \cdot (1 - y)^{k-2} \quad (26)$$

According to Equations 24 and 26, we find that the total number of packets transmitted by DS-TP is:

$$Original + DAR =$$

$$= fs \cdot \textstyle\sum_{k=1}^{n} y^{k-1} \cdot (1 - y)^{k-1} + fs \cdot \sum_{k=1}^{n} y^k \cdot (1 - y)^{k-2}$$

$$= fs \cdot \sum_{k=1}^{n} y^{k-1} \cdot (1 - y)^{k-2} \quad (27)$$

The total transmission delay required by DS-TP is, thus, $DS - TP\ Tran.\ Delay = Original\ Tran.Delay + DAR\, Tran.\ Delay$, or:

$$DS - TP\, Tran.\, D. =$$

$$= \frac{8 \cdot fs \cdot \sum_{k=1}^{n} y^{k-1} \cdot (1 - y)^{k-2}}{1024 \cdot x} \quad (28)$$

Comparison

We divide Equations 16 and 23 by parts, in order to obtain DS-TP's gain against FR-TP, due to DAR:

$$n_{ratio} = \frac{n_{frtp}}{n_{dstp}} = \frac{\frac{\log \frac{1}{fs}}{\log y}}{\frac{\log \frac{1}{fs}}{\log(y \cdot (1-y))}} = 1 + \frac{\log(1-y)}{\log y}. \quad (29)$$

We see that in the current setup the performance difference ratio, in terms of rounds, between the two protocols is totally dependent on the channel packet error rate. We present the performance difference ratio in Figure 7. We observe that for small error rates, the two protocols perform the same (i.e., $n_{ratio} = 1$). As the link error rate increases, DS-TP needs less rounds to complete a file transfer. The performance difference reaches its highest value, when PER = 50%, in which case, DS-TP can complete the file transfer in half as much rounds as FR-TP needs.

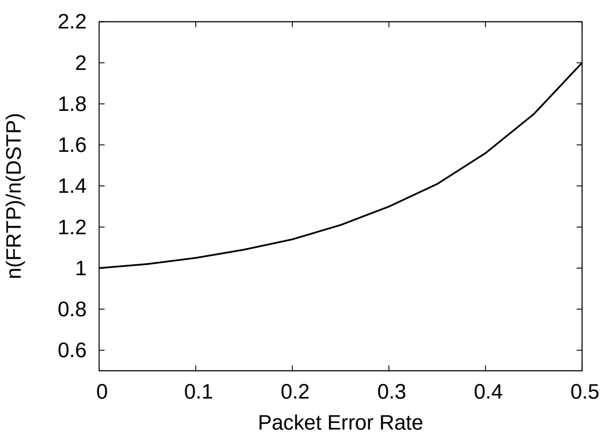


Figure 7. Performance Increase due to DAR in terms of Rounds

In absolute numbers, the difference in rounds between DS-TP and FR-TP is given by $n_{frtp} - n_{dstp}$. Using Equation 23, we arrive at:

$$n_{diff} = n_{frtp} - n_{dstp} = \frac{\log(1-y) \cdot \log \frac{1}{fs}}{\log y \cdot \log(y \cdot (1-y))} \quad (30)$$

In contrast to n_{ratio}, we see that n_{diff} depends, apart from the link error rate, on the file size as well. We present n_{diff} for variable PER in Figure 8 and for variable file size in Figure 9. In Figure 8, we observe that the file transfer can be completed up to 8 rounds faster for DS-TP, than for FR-TP. Obviously, this difference increases even more for larger file sizes. Similarly, in Figure 9 we see that the performance difference increases with the file size. Again, higher PER will favor the performance of DS-TP even more, against FR-TP. Note that in both cases, the performance difference *does not* depend neither on the link speed, nor on the Round Trip Time.

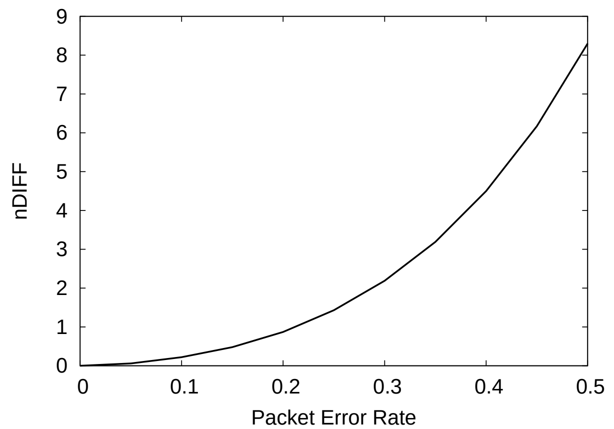


Figure 8. Performance Difference in Terms of Rounds for Variable PER (File Size = 100MB)

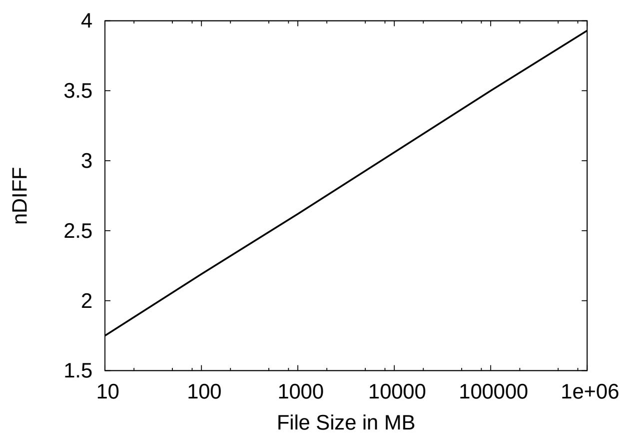


Figure 9. Performance Difference in Terms of Rounds for Variable File Size (PER = 0.3%)

The performance difference in terms of rounds, however, cannot be directly converted to absolute time units, since DS-TP's round is longer than FR-TP's one, due to redundant data transmission. In particular, DS-TP's round is extended for as long as the redundant data take to be transmitted (i.e., DAR Transmission Delay). We use Equation 26 to calculate the *extra* time required by DS-TP in order to complete a round:

$$DAR\,Tr.\,D. = \frac{8 \cdot fs \cdot \sum_{k=1}^{n} y^k \cdot (1-y)^{k-2}}{1024 \cdot x} \quad (31)$$

In order for DS-TP to complete the file transfer faster than FR-TP, the extra transmission delay depicted in Equation 31 should be smaller than the difference, in terms of rounds, captured in Equation 30. In other words, the following Equation should hold:

$$DAR\,Tr.\,Delay \leq n_{diff} \cdot RTT \quad (32)$$

Note, however, that Equation 32 is the worst case scenario for DS-TP. That is, Equation 32 holds when none of the redundant packets sent by DAR fills any holes at the receiver's buffer space (i.e., lost packets belong to $a_{z-1} - r_z$ in Equations 18, 20 *for all* z until the file transfer is complete). In order to include all possible cases, we re-write Equation 32 below using Equations 17 and 28:

$$DSTP\,Tran.\,Del. - FRTP\,Tran.\,Del. \leq n_{diff} \cdot RTT \tag{33}$$

7. CONCLUSIONS

We presented the Deep-Space Transport Protocol (DS-TP), a new protocol for deep space data transfers. DS-TP is an open-loop, rate-based, reliable transport protocol, whose main functionality advantage is incorporated in its efficient and fast retransmission policy. In particular, DS-TP retransmits redundant data packets, in order to fill holes in the receiver's buffer faster than conventional retransmission approaches (i.e., (SN)ACK triggered retransmissions, timeouts, etc.). The redundant packets transmission ratio is set according to the channel packet error rate. Therefore, DS-TP does not inject increased amounts of overhead into the network, in contrast to alternative approaches, such as FEC.

Our theoretical evaluation revealed that DS-TP outperforms conventional transport protocols, similar to Saratoga, by a factor of 2, under specific network conditions (i.e., DS-TP needs half as much time to complete a file transfer). Although we did not present simulation results here, our initial evaluation in a simulation environment reveals that practice (i.e., simulations) follows closely our theoretical analysis. Future work includes, among others i) implementation and comparative evaluation of DS-TP with alternative transport protocols both in simulation and in a real test-bed setup and ii) further investigation and evaluation of the performance of DS-TP when extensive blackout events happen on the transmission link.

REFERENCES

[1] OMNI: Operating Missions as Nodes on the Internet. http://ipinspace.gsfc.nasa.gov

[2] Ian F. Akyildiz, Ozgur B. Akan, Chao Chen, Jian Fang, andWeilian Su, InterPlaNetary Internet: State-of-the-Art and Research Challenges, Computer Networks Journal (Elsevier), Vol. 43, Issue 2, pp. 75-113, October 2003.

[3] R.C. Durst, P.D. Feighery, K.L. Scott, Why not use the Standard Internet Suite for the InterPlaNetary Internet? Available from http://www.ipnsig.org/techinfo.htm.

[4] R.C. Durst, G.J. Miller, E.J. Travis, TCP extensions for space communications, Wireless Networks,Vol. 3, Issue 5, pp. 389-403, October 1997.

[5] O.B. Akan, J. Fang and I.F. Akyildiz, Performance of TCP protocols in deep space communication networks, IEEE Communications Letters, Vol. 6, Issue 11, pp. 478-480, November 2002.

[6] V. Tsaoussidis and I. Matta, "Open issues on TCP for Mobile Computing," The Journal of Wireless Communications and Mobile Computing, WCMC John Wiley and Sons, Vol. 2, Issue 1, pp. 3-20, February 2002.

[7] R. Caceres and L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments", IEEE Journal on Selected Areas in Communications, Vol. 13, Issue. 5, pp. 850-857, June 1995.

[8] I.F. Akyildiz, O.B. Akan and J. Fang, "TP-Planet: A reliable transport protocol for InterPlaNetary Internet," Journal on Selected Areas in Communications, Vol. 22, Issue 2, pp. 348-361, February 2004.

[9] Jian Fang and Ian F. Akyildiz, "RCP-Planet: A Multimedia Rate Control Protocol for InterPlaNetary Internet," International Journal of Satellite Communications and Networking, John Wiley and Sons, Vol. 25, Issue 2, pp. 167-194, February 2007.

[10] Space Communications Protocol Specification (SCPS). Transport Protocol (SCPS-TP). Recommendation for Space Data System Standards, CCSDS 714.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, October 2006.

[11] Consultative Committee for Space Data Systems (CCSDS), Official Site, http://public.ccsds.org/default.aspx.

[12] Lawrence S. Brakmo and Larry L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global Internet," IEEE Journal on Selected Areas in Communications, Vol. 13, Issue 8, pp. 1465-1480, October 1995.

[13] L. Wood, J. McKim, W. Eddy, W. Ivancic and C. Jackson, "Saratoga: A Convergence Layer for Delay Tolerant Networking," work in progress for an -01 internet-draft, submitted to the IETF for the July 2007 meeting.

[14] V. Cerf et al., "Delay-Tolerant Network Architecture," RFC 4838, April 2007.

[15] K. Scott and S. Bureeigh, "Bundle Protocol Specification", IRTF DTNRG Internet Draft revision 10, July 2007.

[16] CCSDS File Delivery Protocol (CFDP). Recommendation for Space Data System Standards, CCSDS 727.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, January 2007.

[17] M. Ramadas et al., "Licklider Transmission Protocol " Specification, work in progress as an IETF internet draft, April 2007.

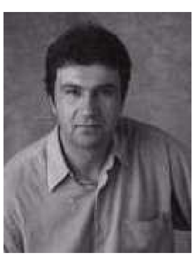
[18] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "Forward Error Correction (FEC) Building Block," RFC 3452, December 2002.

[19] ns 2. The Network Simulator - ns - 2, http://www.isi.edu/nsnam/ns/.

BIOGRAPHY

Ioannis Psaras holds a diploma in Electrical and Computer Engineering from Democritus University of Thrace, Greece in 2004, where he is currently a Ph.D. candidate. He won the Ericsson Award of Excellence in Telecommunications for his diploma dissertation in 2004. Ioannis has worked, as a research intern at DoCoMo Eurolabs (May - September 2005) and at Ericsson Eurolab (May - September 2006). His research interests lie in the area of transport protocols and their behavior over heterogeneous (wired, wireless, satellite, high-speed and deep-space) links.
Personal Web Page: http://utopia.duth.gr/∼ipsaras

Giorgos Papastergiou received his diploma in Electrical and Computer Engineering from Democritus University of Thrace, Greece in 2005. He also received his M.Sc degree in Informatics from Aristotle University of Thessaloniki, Greece in 2007. He is currently working toward the Ph.D degree in the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece. His current research interests are in the area of transport protocols for deep space communication networks.



Vassilis Tsaoussidis received a B.Sc. in Applied Mathematics from Aristotle University, Greece; a Diploma in Statistics and Computer Science from the Hellenic Institute of Statistics; and a Ph.D. in Computer Networks from Humboldt University, Berlin, Germany (1995). Vassilis held faculty positions in Rutgers University, New Brunswick, SUNY Stony Brook and Northeastern University, Boston. In May 2003, he joined the Department of Electrical and Computer Engineering of Democritus University, Greece. His research interests lie in the area of transport/network protocols, i.e. their design aspects and performance evaluation. Vassilis is editor in chief for the Journal of Internet Engineering and editor for the journals IEEE Transactions in Mobile Computing, Computer Networks, Wireless Communications and Mobile Computing, Mobile Multimedia and Parallel Emergent and Distributed Systems. He participated in several Technical Program Committees in his area of expertise, such as INFOCOM, NETWORKING, GLOBECOM, ICCCN, ISCC, EWCN, WLN, and several others.

Nestor Peccia is the Head of the Data Systems Infrastructure Division, Department of Engineering at the Operations Centre of the European Space Agency (ESA / ESOC), in Darmstadt, Germany. Also, he currently serves as the Deputy Chairman CCSDS Engineering Steering Group and Mission Operations and Information Management Services (MOIMS) Area Director and as a Chairman of the European Technology Harmonisation on Space Ground Software Systems.