Computer Communications xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Deep-Space Transport Protocol: A novel transport scheme for Space DTNs

Giorgos Papastergiou, Ioannis Psaras *, Vassilis Tsaoussidis

Dept. of Electrical and Computer Engineering, Democritus University of Thrace, 12 Vas. Sofias Str., 67100 Xanthi, Greece

ARTICLE INFO

Available online xxxx

Keywords:
Deep-space communications
Double Automatic Retransmission
Deep-Space Transport Protocol
High Packet Error Rate

ABSTRACT

The *Delay-/Disruption-Tolerant Networking* Architecture calls for new design principles that will govern data transmission and retransmission scheduling over challenged environments. In that context, novel routing, transport and application layer algorithms have to be established in order to achieve efficient and reliable communication between DTN-nodes.

In this study, we focus on the evolution of the terrestrial Internet into the Interplanetary or Space Internet and propose adoption of the Deep-Space Transport Protocol (DS-TP) as the transport layer scheme of choice for the space networking protocol stack. We present DS-TP's basic design principles and we evaluate its performance both theoretically and experimentally. We verify that practice conforms with theory and observe great performance boost, in terms of file delivery time between DTN-nodes, in case of DS-TP. In particular, the gain of DS-TP against conventional proposals for deep-space communications increases with the link error rate; under conditions DS-TP can improve the performance of the transport layer protocol by a factor of two (i.e., DS-TP can become two times faster than conventional protocols).

1. Introduction

The increasing interest for Space Exploration by the space agencies world-wide has forced the deployment and evolution of data communication networks into the outer space environment. There is common consent among space agencies that telecommunication technologies, such as circuit switched networks, do not perform efficiently in the space environment. Therefore, the Extension of the Internet into Space has become a common goal for scientists and space agencies world-wide. In this context, the *Delay-/Disruption-Tolerant Networking* (DTN) Architecture [6,18] has been proposed to provide an overlay for the Interplanetary or the Space Internet [2,13]. Later, DTN became an interesting idea for challenging environments within the terrestrial Internet as well. For example, DTNs are expected to provide connectivity to the edges of the current Internet infrastructure. In that sense, the DTN Architecture is a potential candidate as an overlay for ad hoc sensor networks, for instance.

However, it is still not clear at all whether common rules can apply for the whole spectrum of *Delay-/Disruption-Tolerant Networks*. For example, reliability guarantees are different for a sensor that gathers temperature samples on the surface of the Earth and a rover/sensor that collects scientific data on the surface of Mars. That said, routing, buffering and congestion avoidance and control issues may exhibit different properties for a terrestrial and a Space DTN. For example, although connectivity may be intermittent in both environments, in a terrestrial DTN once connectivity exists the propagation delay between any two nodes of the DTN will probably be in the order of tens or at most hundreds of milliseconds. In contrast, in a Space or Interplanetary DTN, even when connectivity exists the protocol has to be delay-tolerant, since propagation delays are in the order of tens of minutes. In that context, the goals of a transport layer protocol for Space DTNs are different from the goals of a transport scheme for terrestrial DTNs [8,9,1].

In this paper, we focus on (Deep-) Space or Interplanetary DTNs and evaluate the performance of a novel transport layer scheme, namely the Deep-Space Transport Protocol (DS-TP), which was initially introduced in [15]. DS-TP introduces *proactive transmission and retransmission scheduling rules* in order to deal with the unique characteristics of the deep-space networking environment (e.g., huge propagation delays, high bit error rates, intermittent connectivity, etc.). In particular, DS-TP's basic design principles are based on the fact that deep-space communications are handled, at least presently, by human-operated management procedures that take place long before the mission execution itself. Moreover, DS-TP utilizes the *hop-by-hop, store and forward message switching* principle that governs today's space communications and mitigates the need for congestion avoidance and control. Based on the above, DS-TP transmits data at the *a priori*-known and predetermined line rate. This way, DS-TP achieves high link utilization from the beginning

* Corresponding author. Tel.: +30 25410 79554; fax: +30 25410 79576.
E-mail addresses: gpapaste@ee.duth.gr (G. Papastergiou), ipsaras@ee.duth.gr, ipsaras@gmail.com (I. Psaras), vtsaousi@ee.duth.gr (V. Tsaoussidis).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.02.012

of the file transfer. DS-TP utilizes the functionality of Selective Negative Acknowledgments (SNACKs) in order to signal for holes at the receiver's buffer space. Last but not least, DS-TP's novel, proactive retransmission scheduling policy, called *Double Automatic Retransmission*, allows for efficient and fast retransmission of corrupted data packets. Summarizing, DS-TP can be very well suited as the transport layer scheme of choice for the *Space Delay-/Disruption-Tolerant Networking Stack* [6,18], or alternative schemes like the Delay-Tolerant Transport Protocol (DTTP) [17], or even as the reliability guarantee for the Licklider Transmission Protocol (LTP) [16].

In [15], we introduced the Deep-Space Transport Protocol (DS-TP) and the Double Automatic Retransmission (DAR) technique. We evaluated, theoretically, the properties of DS-TP and compared its performance on a theoretical basis against the Fixed-Rate Transport Protocol (FR-TP). FR-TP is a transport protocol similar, in essence, to the CCSDS File Delivery Protocol (CFDP) [5] and Saratoga [23]. In the present study, we present DS-TP's basic design principles and provide theoretical performance evaluations against FR-TP. We extend our previous study to verify our theoretical results with simulation experiments. Further, we provide numerous simulation results, in order to capture DS-TP's operational properties. We conclude that due to its novel and efficient design principles, DS-TP can complete file transfers faster than conventional transport layer proposals for deep-space communications. However, a number of issues still remain open. For example, we do not elaborate, here, on the end-to-end versus hop-by-hop performance of DS-TP. That is, the increasing number of space objects may allow for alternative transmission paths, dynamic routing schemes and end-to-end transmission of scientific data. Although DS-TP can operate under both scenarios, the comparative performance gain is not evaluated here. Moreover, we do not explore potential Quality of Service guarantees that can be provided by DS-TP.

We organize the rest of the paper as follows: in Section 2, we discuss briefly related proposals for the transport layer of the Space DTN networking stack. In Section 3, we describe in detail the mechanisms and algorithms included in the Deep-Space Transport Protocol. Section 4 includes our Protocol Evaluation Framework, while Section 5 includes the theoretical evaluation of DS-TP versus FR-TP. In Section 6, we verify our theoretical results and further, provide extensive simulation results. Finally, in Section 7, we provide some directions for future research; we conclude the paper in Section 8.

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.

One of the early proposals for reliable data transmission over deep space links is TP-Planet [3]. 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 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 execution. Therefore, congestion control is not really needed, since flow multiplexing over deep-space links does not exist, at least presently. 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 [10] 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 include 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) [19] is a protocol developed by the Consultative Committee for Space Data Systems (CCSDS) [7] 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 [4] 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 [23] 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 the 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 and presumes (strong) link layer (FEC) coding. Saratoga can be used as a convergence layer to exchange Delay-Tolerant Networking bundles [6,18] between peer nodes. To the best of our knowledge, Saratoga is the first protocol evaluated on a real satellite testbed [22].

A similar file-oriented protocol is the CCSDS File Delivery Protocol (CFDP) [5], which is mainly an application layer protocol that includes 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 Functionality) 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, CFDP presumes link level coding.

Similarly to Saratoga, the Licklider Transmission Protocol (LTP) [16] 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.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012

Lately, *erasure coding* has attracted some attention in the context of data transmission over challenged networks. For example, authors in [20] apply erasure coding techniques to optimize routing performance in delay-tolerant networks. They argue that current trends towards redundant-based transmissions result in either high overhead or long delays due to wrong retransmission choices. On the same direction, authors in [21] compare erasure coding techniques with replication-based mechanisms. They conclude that erasure coding improves the system's performance by several orders of magnitude. Although both of the above studies are comprehensive and produce sound evidence regarding the superiority of erasure coding against replication based techniques, *they both refer to terrestrial DTNs or DTNs, where once connectivity exists, propagation delays are minimal*. In contrast, in the deep-space communication environment, delays are extremely high, even in case of end-to-end or point-to-point connectivity. To that end, we consider that further research is needed in order to conclude whether the above conclusions hold for the space networking environment as well.

Moreover, sophisticated FEC-based techniques such as FLUTE [14] have the potential to improve the system's performance; see for example [11], where the authors propose an architecture to improve DTN communication in sparsely populated areas. Uni-DTN is a unidirectional DTN convergence layer, which can provide scalability for unicast and multicast distribution of DTN bundles. Although the ideas included in[14,11] seem to be promising, they target different communication environments and therefore, comparisons cannot be made easily; we include these approaches here for completeness. Future research may uncover whether techniques included in those approaches can be included in our proposal or vice versa (e.g., the ACK–SNACK approach adopted in DS-TP could be integrated into an extension of the Uni-DTN architecture).

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.

3.1. Basic components

DS-TP is based on the following fundamental characteristics:

(1) *Rate-based transmission.* The huge propagation delays experienced in deep-space communications prohibit the real-time discovery of transmission link bandwidths. Therefore, the bandwidths of the transmission links as well as the intervals during which those links are available have to be announced in advance. That said, a deep-space DTN entity is aware of the available bandwidth that it can utilize, at any given point in time. In that context, a fixed-rate transmission tactic allows for high link utilization, without forcing transmission rate increase, which by definition leads to congestion losses (at least in the long term).

(2) *Mixed ACK–SNACK Strategy*. Clearly, the above situation calls for decoupling of the ACK role from transmission rate adjustments. 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; in contrast, SNACKs are used as a complementary mechanism to either allow for network measurements or trigger retransmission of lost segments. The detailed operation and functionality of SNACKs is presented later on in this section.

(3) *Double Automatic Retransmission.* 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 conventional retransmission approaches (e.g., TCP-RTO). 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 can be initially set to a small value that corresponds to high PER (e.g., 50%) and can be adjusted according to the actual PER, based on network measurements. Alternatively, R_d can be set according to estimations for the link error rate, in order to avoid high overhead at the beginning of the transfer. We consider, however, that this is an implementation-specific issue and therefore, we do not elaborate further on that in the present study. DAR is presented in detail below.

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

3.2. Double Automatic Retransmission (DAR)

3.2.1. Transmission sequence

As we have already mentioned earlier, DS-TP injects data packets into the transmission link in a predetermined, fixed rate (i.e., the *Actual Rate*). Apart from the *Actual Rate*, the DS-TP sender keeps one extra variable, called *Retransmission Rate* and referred to as R_Ratio, 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 simplicity, in this work we assume that each loss occurs independently with probability p. 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 Fig. 1). In other words, one redundant packet is transmitted every $\frac{1}{error\_rate} - 1$ original packets. Although this is a subject that calls for further investigation, as we show in Sections 6 and 7, in the current setup DS-TP is designed

Table 1
Symbols.

Symbol	Meaning
c_seqno	Current sequence number
r_seqno	Retransmission sequence number
x	Link transmission rate
fs	File size
error_rate	Link error rate
y	*error_rate*
R_Ratio	DAR retransmission rate

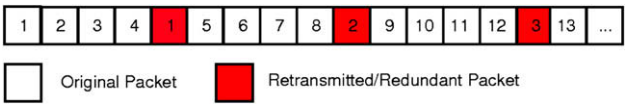


Fig. 1. Example packet transmission sequence.

and evaluated according to the above. The retransmission sequence number is, thus, given by:

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

Obviously, whenever Eq. (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:

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

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.

Summarizing, if *error_rate* = 20%, the packet with `c_seqno` = 3 will be retransmitted after 12 packets, according to Eq. (2), since at that time `r_seqno` = 0, according to Eq. (1). The packet transmission and retransmission sequence, in that case, is shown in Fig. 1.

In other words, DAR transmits redundant packets with *error_rate* or R_Ratio Mbps. The original-packet transmission rate is, thus, reduced to *Original Packet Rate* = *Link Rate* − *error_rate*. Referring to the previous example, we have that *Original Packet Rate* = 80% · *Link Rate*.[1]

3.2.2. 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, to acknowledge the packets that have successfully arrived 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 Eq. (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 $SNACK_1$. 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., Eq. (1)), the DS-TP receiver expects arrival of the redundant packet, according to Eq. (1). In case the redundant packet does not arrive, within that interval, which we call *DAR_intr*, a $SNACK_2$ is sent. In contrast to $SNACK_1$, $SNACK_2$ triggers immediate retransmission of the missing segment(s).

There is, however, one salient point in the above operation, which we attempt to address briefly below. Each redundant packet is transmitted by the sender (and consequently arrives at the receiver) after *diff_pkts* number of packets. Depending on the link's transmission delay, the redundant packet arrives at the receiver side after *diff_time* time-units. Obviously, *diff_time* = *DAR_intr*. In order for the DAR retransmission to be faster than a $SNACK_2$ triggered retransmission, *diff_time* has to be shorter than the Propagation Delay of the Reverse channel, according to:

$$diff\_time \leqslant Reverse\ Channel\ Prop.\ Delay \quad (3)$$

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 (i) the current sequence number, (ii) the (predetermined) link speed and (iii) the consequent transmission delay; based on (i–iii) the DS-TP sender schedules the retransmission attempts accordingly. In particular, if $diff\_time \leqslant \frac{RTT}{2}$, 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. Due to the predetermined nature of deep-space communications, the sender is able to calculate the current sequence number (i.e., *c_seqno*) boundary that cancels DAR. According to that, the sender triggers immediate retransmission upon $SNACK_1$ arrival. We refer the reader to [15] for an extended discussion on that issue.

Therefore, for sequence numbers greater than the calculated *c_seqno* boundary, 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`:

$$r\_seqno \leftarrow c\_seqno. \quad (4)$$

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 Eq. (4). However, the previous analysis needs to be extended in order to include this shift in *r_seqno*. Apart from the *c_seqno* and *r_seqno* variables, the sender maintains one more variable, namely the retransmission *jump sequence number* variable (*j_seqno*). *j_seqno* holds the last shift in *r_seqno*, namely the value of *c_seqno* at the time when Eq. (3) does not apply anymore. Taking into account *j_seqno* Eqs. (1) and (2) become:

$$r\_seqno = \frac{(c\_seqno - j\_seqno) - 1}{\frac{1}{error\_rate} - 1} + j\_seqno, \quad (5)$$

and

$$diff\_pkts = \left[\left(\frac{1}{error\_rate} - 1\right) \cdot (c\_seqno - j\_seqno)\right] - (r\_seqno - j\_seqno). \quad (6)$$

[1] 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.

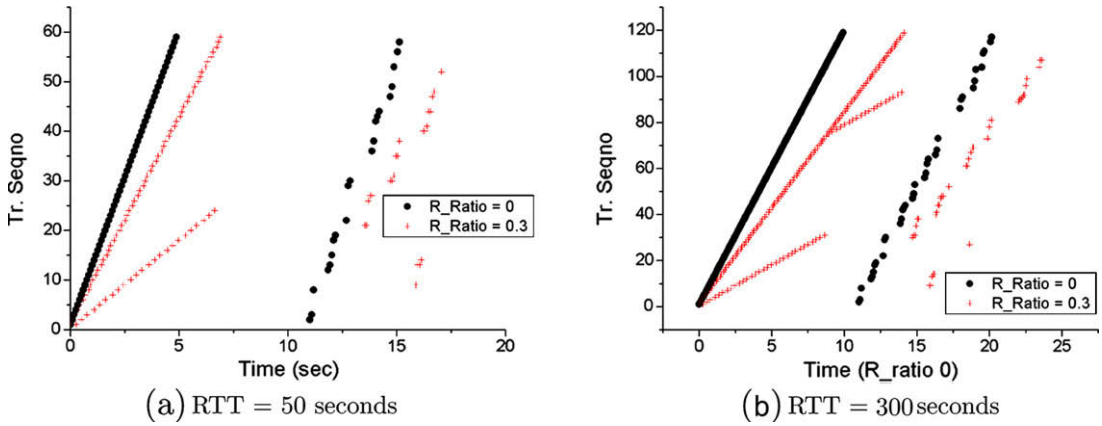


Fig. 2. File size = 4×BDP.

We depict the above operation in Fig. 2. The black dots in Fig. 2 represent the transmission sequence of DS-TP when R_Ratio = 0, while the grey crosses ("+") capture DS-TP's transmission and retransmission sequence when R_Ratio = 0.3. In Fig. 2(a), we see that DAR retransmits data packets according to its operational rules, but in this case there is no need to shift the *r_seqno* during the first round. In contrast, in Fig. 2(b), we see that *r_seqno* needs to be shifted from sequence number 29 to sequence number 79. This means that no packet in between 29 and 79 is sent twice by DAR and that corrupted packets within that interval are retransmitted by $SNACK_1$ arrival at the sender side.

In order to deal with corrupted packets, whose retransmission is triggered by $SNACK_2$, the receiver schedules a timer for each transmitted SNACK2. Upon the timer's expiration, the DS-TP receiver sends a second $SNACK_2$ in order to trigger retransmission of lost packets.

The retransmission timeout value (for $SNACK_2$) is set approximately equal to the path RTT. Further investigation is needed in order to choose a dynamically adjustable timeout value, although the present scheme performs pretty efficiently. Moreover, sender-oriented retransmission timers for that purpose have been evaluated during our study as well. We report, however, that such approaches do not perform efficiently within the context of DS-TP, due to its open-loop transmission tactic.

Finally, we note that due to congestion or corruption on the reverse path, ACKs or SNACKs may get lost. However, we do not apply yet any DAR-like technique on the reverse path. We assume that retransmitting (SN)ACKs back-to-back multiple (two or three) times will mitigate this problem. We leave this issue as a subject of future work.

4. Protocol evaluation framework

In Section 5, we attempt to comparatively evaluate, on a theoretical basisf, the performance of a modified, simpler version of DS-TP with a protocol, whose functionality is very close to that of CFDP [5] and Saratoga [23]. We consider that the *Fixed-Rate Transport Protocol* (FR-TP) can capture with acceptable accuracy the transmission performance of protocols like Saratoga [23,22] or CFDP [5]. 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. Next, in Section 6, we evaluate the performance of the two protocols using simulation experiments. We implemented both DS-TP and the lightweight alternatives of similar proposals, such as Saratoga [23] and CFDP [5], in the ns-2 [12] simulation platform.

In particular, we consider the *Fixed-Rate Transport Protocol* (FR-TP), whose main functionality is summarized as follows: the FR-TP sender sends data at 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 holes in the incoming transmission sequence. To simplify the theoretical analysis, we consider that SNACKs are sent back to the sender, only after the whole transmission attempt has already completed. This operation is similar to the deferred mode of CFDP [5].

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., Section 3), apart from its SNACK triggered retransmission policy. In particular, the DS-TP receiver sends SNACKs for missing packets after completion of the whole transmission attempt, similarly to FR-TP.[2] We note that in our experimental evaluation the assumption of late SNACK transmission is relaxed. We evaluate the protocols' dynamics under such scenarios in Section 6, through simulations.

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 [18]. Obviously, the primary 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 and link utilization boundaries.

5. DS-TP versus 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*.

[2] Alternatively, one may consider that the file size is smaller than the capacity of the forward channel, so that feedback arrives after the sender has already transmitted the whole file.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012

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 packets has to be transferred across the deep-space link to the receiver.

5.1. FR-TP

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

$$fs \cdot y^n < 1 \ packet. \tag{7}$$

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

$$n_{frtp} = \log_y(y^n) = \log_y\left(\frac{1}{fs}\right) = \frac{\log\frac{1}{fs}}{\log y}. \tag{8}$$

Whenever the above equation leads to a non-integer value for n, n is rounded upwards.

5.2. 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 $fs + r_1$ packets, where r_1 is the number of retransmitted packets during this round. In order to depict the retransmission overhead with respect to the file size, we use and modify Eq. (1) as follows:

$$retr = \frac{fs - 1}{\frac{1}{error\_rate} - 1} \simeq \frac{fs}{\frac{1}{error\_rate} - 1} = \frac{fs \cdot error\_rate}{1 - error\_rate}. \tag{9}$$

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 $r_1 = fs \cdot \frac{y}{1-y}$ (Eq. (9)). 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:

$$a_1 = (fs - r_1) \cdot y + r_1 \cdot y^2. \tag{10}$$

Substituting r_1 into Eq. (10), we get that:

$$a_1 = fs \cdot y \cdot (1 - y). \tag{11}$$

Similarly, during the second round, where a_1 packets 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:

$$a_2 = (a_1 - r_2) \cdot y + r_2 \cdot y^2. \tag{12}$$

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 current implementation differs from the above assumption, the result remains intact in both cases. In particular, DS-TP's current implementation applies DAR: (i) for the initial file transmission, as described above, and (ii) for each block of packets signaled by each $SNACK_2$, independently of the packets signaled by other $SNACK_2$s. We elaborate further on the practical implementation of DS-TP in Section 6. In all cases, our theoretical evaluation can very well represent the "Deferred" or "Prompted" ACK strategy adopted in [5] or [23].

Substituting r_2 into Eq. (10), we get that:

$$a_2 = fs \cdot y^2 \cdot (1 - y)^2. \tag{13}$$

DS-TP will complete the file transfer, when $a_z < 1$, where $z = n - 1$. Generalizing Eqs. (11) and (13), we assume that the file transfer is complete, once the following equation holds:

$$fs \cdot y^n \cdot (1 - y)^n < 1 \ packet. \tag{14}$$

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

$$n_{dstp} = \log_{[y\cdot(1-y)]}[y \cdot (1 - y)]^n = \log_{[y\cdot(1-y)]}\left(\frac{1}{fs}\right) = \frac{\log\frac{1}{fs}}{\log(y \cdot (1 - y))}. \tag{15}$$

Note that Eq. (14) 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 Eq. (14) as follows, and we call these packets *Original*:

$$Original = fs \cdot y^{n-1} \cdot (1 - y)^{n-1}. \tag{16}$$

In [15], we include extensive calculations for the transmission delay of both DS-TP and FR-TP.

5.3. Comparison

We divide Eqs. (8) and (15) by parts, in order to obtain DS-TP's gain against FR-TP, due to DAR:

$$n_{ratio} = \frac{n_{frtp}}{n_{dstp}} = \frac{\frac{\log\frac{1}{fs}}{\log y}}{\frac{\log\frac{1}{fs}}{\log(y\cdot(1-y))}} = 1 + \frac{\log(1 - y)}{\log y}. \tag{17}$$

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 Fig. 3. 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 many rounds as FR-TP needs.

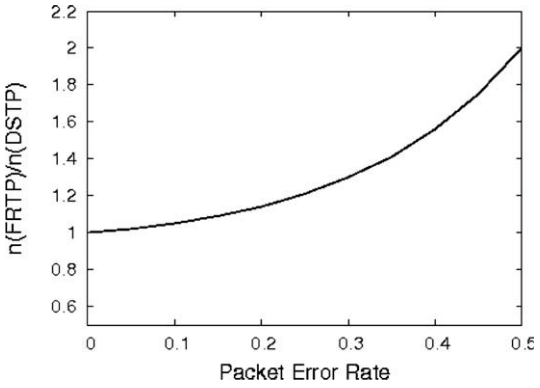
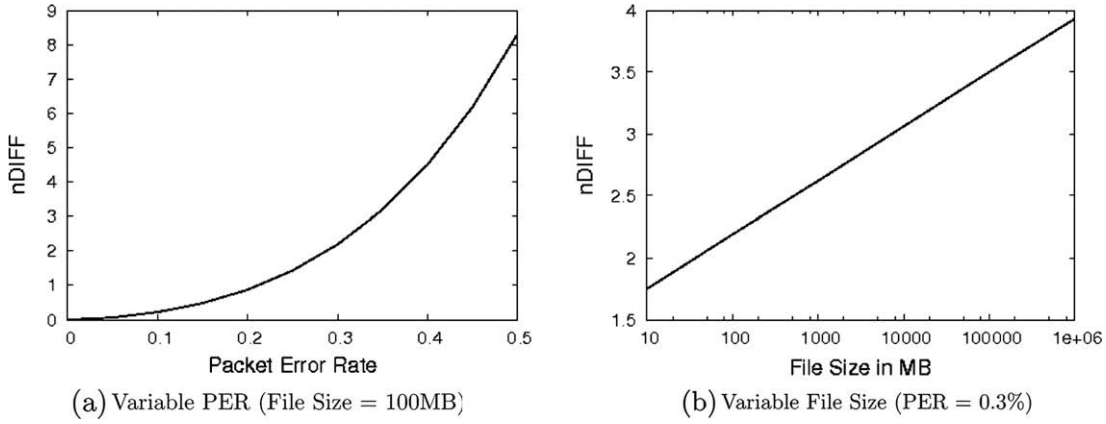


Fig. 3. Performance increase due to DAR in terms of rounds.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012



(a) Variable PER (File Size = 100MB) (b) Variable File Size (PER = 0.3%)

Fig. 4. Performance difference 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 Eq. (15), we arrive at:

$$n_{diff} = n_{frtp} - n_{dstp} = \frac{\log(1-y)\cdot \log\frac{1}{fs}}{\log y \cdot \log(y\cdot(1-y))}. \quad (18)$$

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 Fig. 4(a) and for variable file size in Fig. 4(b). In Fig. 4(a), 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 Fig. 4(b) 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 depends *neither* on the link speed, *nor* on the Round Trip Time.

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 it takes for the redundant data to be transmitted (i.e., *DAR Transmission Delay*). We refer the reader to [15] for transmission delay calculations. There, we calculate the total file transmission delay both for DS-TP and for FR-TP and we provide the corresponding inequality that should hold in order for DS-TP to outperform FR-TP. In Section 6, we show that from a point onwards, when the file size increases, the extra transmission delay induced by DAR results in extensive file delivery time.

6. Simulation results

6.1. Simulation setup

We have implemented and evaluated the performance of DS-TP on the ns-2 network simulator [12]. For the purpose of the present study, we decouple the DAR retransmission rate from the link error rate measurement; our goal is to identify the most appropriate combination between the *link error rate* and *DAR retransmission ratio* (R_Ratio); we evaluate the performance of DS-TP according to that combination. The length of the SNACK Bit Vector is fixed for all experiments and is equal to 1 Byte. That is, the number of packets that can be (SN)ACKed by each SNACK segment is eight packets plus the size, in terms of packets, of the first hole in the corresponding receiver's buffer space. Since the SNACK Bit Vector length is common for both DS-TP and FR-TP, we consider the comparison fair; the optimal SNACK Bit Vector length is left as a subject of future work, since it needs to be investigated in conjunction with channel asymmetries, which are not considered here. The timer that triggers retransmission of a $SNACK_2$ (i.e., in case a packet, whose retransmission is triggered by a previous $SNACK_2$ is lost) is set equal to the Round Trip Time (RTT) of the link plus some extra delay in order to account for the transmission delay of the retransmitted packets.

Table 2
Simulation parameters.

Parameter	Value
Link rate (Mbps)	1
One-way propagation delay (s)	25, 150
Packet size (Bytes)	1050
File size (percentage of bandwidth-delay product)	0.25, 0.5, 1, 2, 4
Packet error rate	0.05, 0.1, 0.2, 0.3, 0.4, 0.45, 0.5
R_Ratio	0.0, 0.1, 0.2, 0.3, 0.4, 0.45, 0.5

We evaluate the performance of DS-TP and FR-TP over a point-to-point deep-space link; we perform simulations for varying distance and link error rate and thus, for varying bandwidth-delay product as well. The protocols' performance is evaluated in terms of the required File Delivery Time for 490 combinations of file size, packet error rate, propagation delay and DAR retransmission rate. For each combination of the above parameters, we repeat the simulation with 20 different seeds and we present their average outcome with 95% confidence interval. In Table 2, we present the configuration of our simulation setup. We used the embedded ns-2 Bernoulli error model, in order to simulate the link error rate.

6.2. Theoretical analysis verification

In this Section, we verify our theoretical results with simulations, presented in Section 5. The file size for the simulations presented in this Section is set to 0.25 × BDP, although similar results were obtained for larger file sizes, as we show later on in this Section. The retransmission rate of DAR is set equal to the link PER, according to DS-TP's initial design principles. Note that throughout our simulations, the link PER is kept constant in order to draw some initial conclusions regarding the performance of DAR and DS-TP. In Figs. 5 and 6, we compare our theoretical results (Figs. 3 and 4) with the simulations' outcome for RTT = 50 s and RTT = 300 s, respectively. We explicitly note at this point that we have chosen relatively small values for the round-trip propagation delay of the deep-space link, in order to speed-up the required simulation time. However, although such values may not represent precisely the propagation delay of a deep-space link, results are commensurate for larger RTT values as well. This can be further explained by the fact that our chosen values for the simulated file

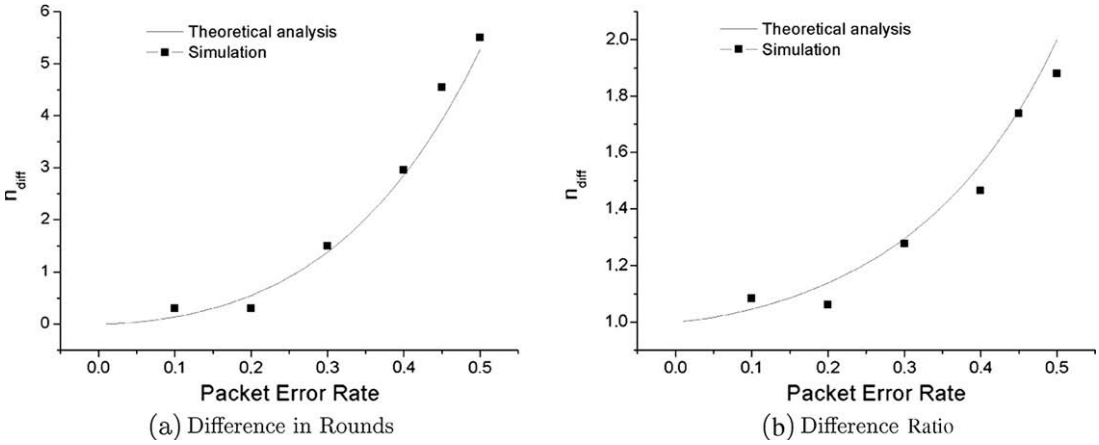


Fig. 5. Performance difference, RTT = 50 s.

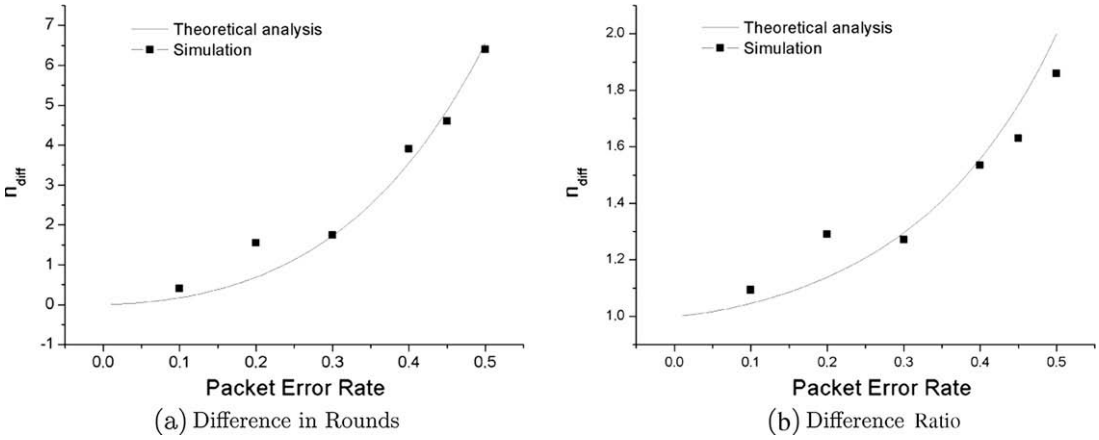


Fig. 6. Performance difference, RTT = 300 s.

sizes are set as a percentage of the Bandwidth-Delay Product (BDP). Therefore, as the Delay increases, the BDP and consequently the file size increase as well. Thus, results follow the same patterns in all cases (compare, for example, Figs. 5 and 6).

In all cases (i.e., Figs. 5 and 6), we observe excellent agreement between theoretical analysis and simulation results. We observe that the difference, in terms of rounds (Figs. 5(a) and 6(a)), is slightly greater when RTT = 300 s, since the file size is larger in that case and thus, DAR's operational benefits can be better exploited. Based on the above, we conclude that practice (i.e., simulations) follows closely theory.

6.3. Performance evaluation

In this Section, we extend our evaluation to include also larger file sizes. In particular, we simulate file sizes equal to 0.25 × BDP,

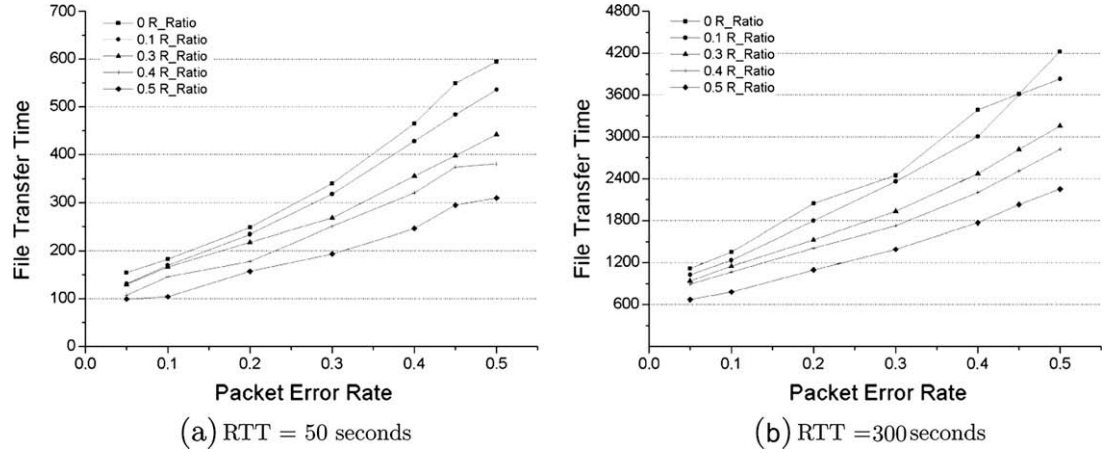


Fig. 7. File size = 0.25 × BDP.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012

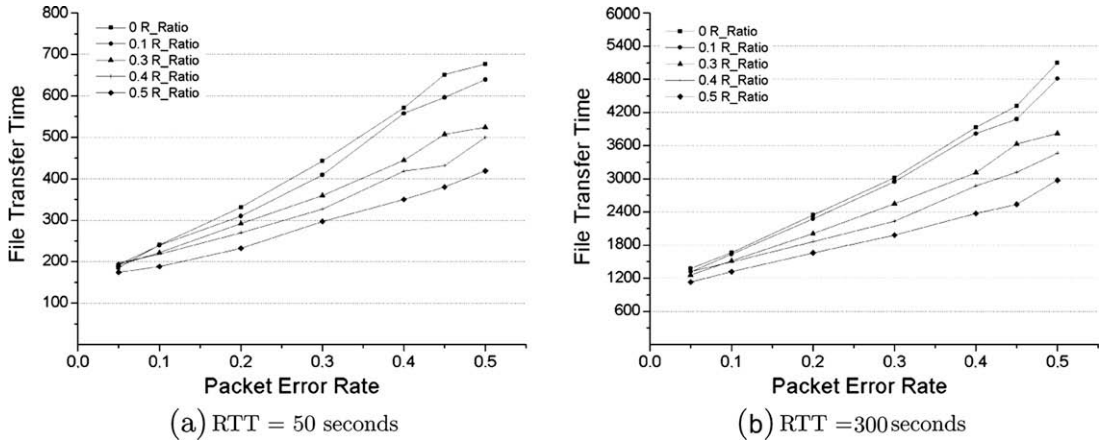


Fig. 8. File size = 1 × BDP.

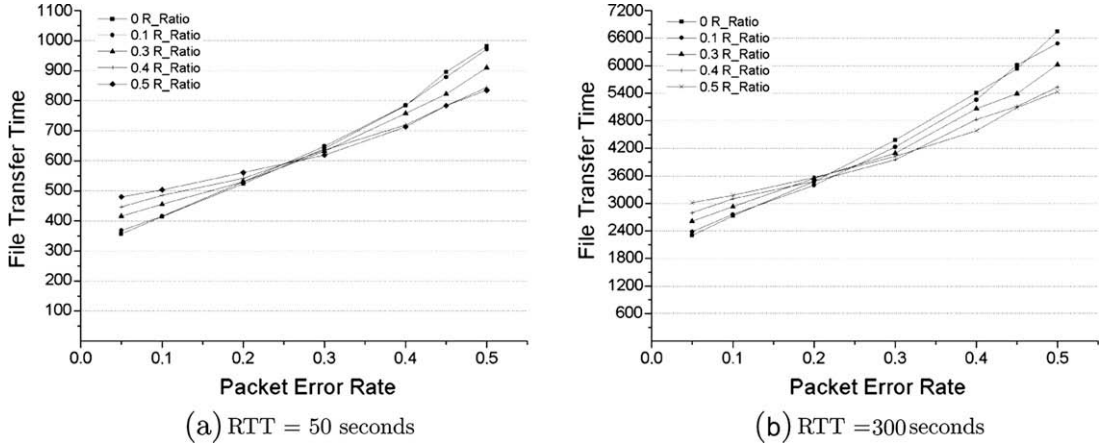


Fig. 9. File size = 4 × BDP.

1 × BDP and 4 × BDP. The simulations' outcome is presented in Figs. 7–9. Each one of the lineplots in Figs. 7–9 refers to a specific DAR retransmission rate. The rates range from R_Ratio = 0 (i.e., FR-TP) to R_Ratio = 0.5. We observe the following:

(1) *The gain of the Double Automatic Retransmission and consequently of DS-TP as well, depends (i) on the file size and (ii) on the link error rate.* In particular, the smaller the file size (normalized here over the BDP of the deep-space link) and the higher the link error rate, the higher the gain of DS-TP against FR-TP. For example, in Fig. 7(a), when PER = 5% and R_Ratio = 0.5, DS-TP gains one round (i.e., 50 s). In contrast, for the same link error rate and R_Ratio, in Fig. 8(a), we see that DS-TP gains less than one round. The situation is similar for larger end-to-end propagation delays (see Figs. 7(b) and 8(b)). By the same token, when the link error rate increases, the gain of DS-TP increases as well. In particular, for high error rates DS-TP is twice as fast as FR-TP is (see, for example, the case of 50% PER in Figs. 7 and 8).

(2) *The file transfer time, when normalized to the RTT, is more dependent on link PER than to the file size.* We see in Fig. 8(a) that when R_Ratio = 0.5 and for link error rates ranging from 5% to 50% the file transfer time raises from 3.48 to 8.37 RTTs. On the other hand, comparing Figs. 8(a) and (b) when PER = 50% and R_Ratio = 0.5, the file completion time raises from 8.37 to 9.9 RTTs. For lower DAR rates this difference becomes even smaller.

(3) *As the file size increases, FR-TP may become more efficient.* We observe, in Fig. 9, that for small link error rates, FR-TP achieves faster file transfer. However, as the error rate increases, DS-TP becomes faster again. In particular, the total number of RTTs gained due to DAR faster retransmission is less than the additional transmission delay introduced by the DAR redundant packet transmission. Indeed, we see in Fig. 10, where the R_Ratio = 0.5, that when the file size becomes greater than two times the BDP of the transmission link, the file delivery time increases to prohibitive levels. We refer the reader to [15] for a comprehensive analysis of the required transmission delay for FR-TP and DS-TP. We consider, however, that due to the huge capacity of deep-space links, the file size will rarely be as large as 4 × BDP.

(4) *There exists a tradeoff between the optimal R_Ratio and the extra overhead introduced by DAR.* In Figs. 7–9, we see that DS-TP is always faster when R_Ratio = 0.5. This value for the R_Ratio, however, introduces high retransmission overhead and extra transmission delay for DS-TP. In the present study, we do not explore the optimal combination between the link error rate and the DAR retransmission rate (i.e., R_Ratio). On that direction, we present performance evaluation results in terms of file delivery time, for a variety of file sizes, when R_Ratio = 0.5 (see Fig. 10). Indeed, we see that when the file size increases, high values for the R_Ratio lead to extensive transmission delays, which in turn increase the overall file delivery time.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012

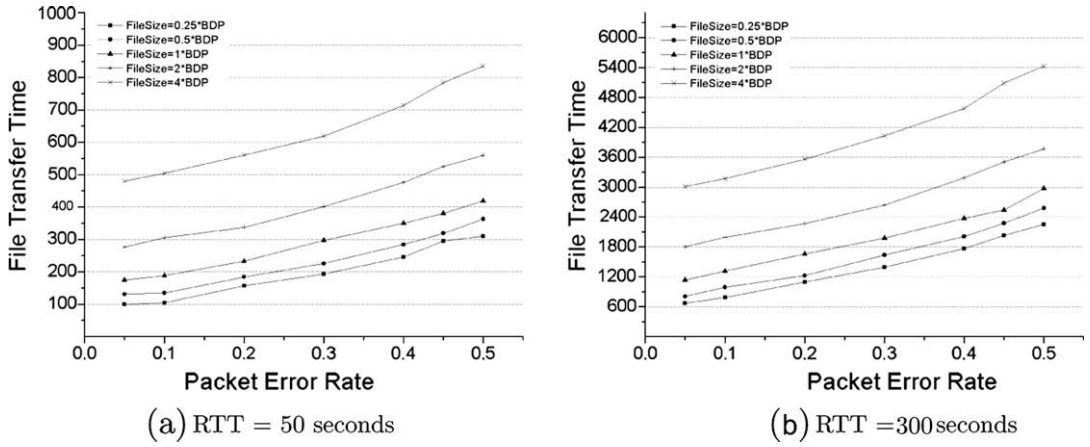


Fig. 10. R_Ratio = 0.5.

A closer look to Figs. 7 and 8 reveals that for small files sizes (e.g., 0.25 × BDP), where the transmission delay is relatively small and for the cases where the retransmission rate is equal to the link PER, DS-TP's performance gain matches well the theoretical analysis (i.e., Figs. 5 and 6), which did not include the transmission delay of the file. This implies that an analysis similar to the one presented in Section 5 that would account for different combinations of link PER and DAR-rate would accurately predict the performance of DS-TP for small file sizes with regard to the BDP.

7. Open issues and future work

We have presented and evaluated DS-TP's core ideas and functionalities. Although the theoretical evaluation and the simulation results presented here give a good insight of the protocol's inherent properties, some issues need to be further investigated and evaluated. In this Section, we address some of the open issues identified during the protocol's implementation and give some directions for future work.

The theoretical evaluation presented in this work compares the performance of DS-TP with the so-called Fixed-Rate Transport Protocol (FR-TP). However, our model applies only to the case where DAR rate is set equal to the measured link error rate and does not consider the additional delay inserted due to DAR retransmissions. Furthermore, it assumes transmission of one type of SNACKs which takes place at the end of each round and thus, simplifies DAR's functionality. In this context, we plan to extend our model so as to accurately predict DS-TP's file delivery time, due to extra transmission delays [15], and use this model to exploit better the tradeoff between file delivery time and (i) retransmission overhead, (ii) extra transmission delay due to redundant transmission.

Furthermore, there is a number of open issues, related to DS-TP's operation, which affect its behavior performance-wise. For example, deviations in error rate estimation between the sender and the receiver may cause the DAR mechanism to misfunction. Moreover, performance implications of the SNACK bit-vector length have not been investigated here.

In the present study, we have not considered any DAR-like techniques for SNACK transmissions at the reverse direction, or for retransmissions triggered by $SNACK_2$s at the sender side. These issues will be defined and implemented as part of our future work.

Finally, the potential benefits of DAR against intermittently connected or highly partitioned environments need to be further investigated. DS-TP's current implementation does not include any specific mechanism to deal with blackouts, but rather assumes that packet corruption follows the standard ns-2 Bernoulli distribution. Clearly, long-lasting blackouts, or burst losses which will become common with the increasing adoption of Ka-band transmission on deep-space links will result to false error rate estimations; in turn, a large number of corrupted or lost SNACKs will need to be retransmitted after the timers' expirations. Moreover, this will increase the value for the R_Ratio, which means that DS-TP may induce high overhead (and consequently lower bandwidth utilization) unnecessarily. Thus, a mechanism specifically designed to deal with these conditions should be incorporated within DS-TP's core functionality.

8. Conclusions

The *Deep-Space Transport Protocol* (DS-TP) is a novel approach to reliable data transmission over deep-space transmission links. The *Double Automatic Retransmission* (DAR) technique suffices to utilize the huge capacity of the deep-space link and moreover, exploit its capacity for redundant data (re-) transmission in order to account for high corruption rates. That is, retransmission attempts are scheduled *proactively*: redundant data packets are multiplexed with original packets in order to avoid corruption of both the original and the redundant data packet.

Our theoretical analysis, presented also in [15], revealed that DS-TP can complete file transfers faster than conventional transport protocols for deep-space communications. In the present study, we verified our theoretical results with simulation experiments and extended our evaluation to include additional scenarios. Although several issues still remain open, the present study provides a good insight regarding the performance of DS-TP. Thus, we consider that DS-TP presents high potential for deployability.

References

[1] O.B. Akan, J. Fang, I.F. Akyildiz, Performance of TCP protocols in deep space communication networks, IEEE Communications Letters 6 (11) (2002) 478–480.
[2] Ian F. Akyildiz, Ozgur B. Akan, Chao Chen, Jian Fang, Weilian Su, InterPlaNetary Internet: state-of-the-art and research challenges, Computer Networks Journal 43 (2) (2003) 75–113.
[3] I.F. Akyildiz, O.B. Akan, J. Fang, TP-Planet: a reliable transport protocol for InterPlaNetary Internet, Journal on Selected Areas in Communications 22 (2) (2004) 348–361.
[4] Lawrence S. Brakmo, Larry L. Peterson, TCP Vegas: end-to-end congestion avoidance on a global Internet, IEEE Journal on Selected Areas in Communications 13 (8) (1995) 1465–1480.
[5] CCSDS File Delivery Protocol (CFDP), Recommendation for Space Data System Standards, CCSDS 727.0-B-4, Blue Book, CCSDS, Washington, DC, Issue 4, January 2007.
[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, Delay-Tolerant Network Architecture, RFC 4838, April 2007.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012

[7] Consultative Committee for Space Data Systems (CCSDS), Official Site. Available from: <http://public.ccsds.org/default.aspx/>.
[8] 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/>.
[9] R.C. Durst, G.J. Miller, E.J. Travis, TCP extensions for space communications, Wireless Networks 3 (5) (1997) 389–403.
[10] Jian Fang, Ian F. Akyildiz, RCP-Planet: a multimedia rate control protocol for InterPlaNetary Internet, International Journal of Satellite Communications and Networking 25 (2) (2007) 167–194.
[11] Dirk Kutscher, Janico Greifenberg, Kevin Loos, Scalable DTN distribution over Uni-Directional Links, in: Proceedings of NSDR'07, Kyoto, Japan, August 27, 2007.
[12] ns 2, The Network Simulator-ns-2. Available from: <http://www.isi.edu/nsnam/ns/>.
[13] OMNI: Operating Missions as Nodes on the Internet. Available from: <http://ipinspace.gsfc.nasa.gov/>.
[14] T. Paila, M. Luby, R. Lehtonen, V. Roca, R. Walsh, FLUTE: File Delivery over Unidirectional Transport, RFC 3926, October 2004.
[15] Ioannis Psaras, Giorgos Papastergiou, Vassilis Tsaoussidis, Nestor Peccia, DS-TP: Deep-Space Transport Protocol, in: Proceedings of IEEE Aerospace Conference, March, Montana, USA, 2008.
[16] M. Ramadas, S. Burleigh, S. Farrell, Licklider, Transmission Protocol-Specification, RFC 5326, experimental, September 2008.
[17] Christos V. Samaras, Vassilis Tsaoussidis, DTTP: a Delay-Tolerant Transport Protocol for Space Internetworks, ERCIM, 2008.
[18] K. Scott, S. Burleigh, Bundle Protocol Specification, RFC 5050, November 2007.
[19] Space Communications Protocol Specification (SCPS), Transport Protocol (SCPS-TP). Recommendation for Space Data System Standards, CCSDS 714.0-B-2, Blue Book, CCSDS, Washington, DC, Issue 2, October 2006.
[20] Yong Wang, Sushant Jain, Margaret Martonosi, Kevin Fall, erasure-coding based routing for opportunistic networks, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, pp. 229–236.
[21] Hakim Weatherspoon, John Kubiatowicz, Erasure coding vs. replication: a quantitative comparison, in: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 02), vol. 78, MIT Faculty Club, Cambridge, MA, USA, March 2002.
[22] L. Wood et al., Use of Delay-Tolerant Networking Bundle Protocol from Space, IAC 2008, B2.3.10. Available from: <http://personal.ee.surrey.ac.uk/Personal/L.Wood/publications/wood-dtn-bundle-space-IAC-08.B2.3.10.pdf/>.
[23] L. Wood, J. McKim, W. Eddy, W. Ivancic, 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.

Please cite this article in press as: G. Papastergiou et al., Deep-Space Transport Protocol: A novel transport scheme for Space DTNs, Comput. Commun. (2009), doi:10.1016/j.comcom.2009.02.012