

# DTTP: a Delay-Tolerant Transport Protocol for Space Internetworks

Christos V. Samaras<sup>1</sup>, Vassilis Tsaoussidis<sup>1</sup> and Nestor Peccia<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering  
Democritus University of Thrace  
12 Vas. Sofias Str., Xanthi 67100, Greece  
{csamaras, vtsaousi}@ee.duth.gr

<sup>2</sup>ESA/ESOC  
Robert Bosch Strasse 5, D-64293 Darmstadt, Germany  
nestor.peccia@esa.int

**Abstract.** We propose Delay-Tolerant Transport Protocol (DTTP) to address reliable data transfer in stressed network environments, such as space communications. Since existing TCP mechanisms do not work well (or at all) for such networks, new transport schemes are required. Intermittent connectivity in space environments calls for new transport approaches that smoothly adapt to the special networking conditions. DTTP is primarily a transport layer protocol and satisfies the inherent architecture requirements of Delay Tolerant Networking (DTN) in the absence of IP network infrastructure. It allows for reliable, efficient data transfer offering a number of application-oriented transmission strategies. Otherwise, when an IP architecture exists, DTTP operates as a stand-alone transport entity which interfaces with IP directly. We introduce the protocol's properties and functionality that enable its deployment in challenged networks. We conduct simulations that demonstrate the protocol's efficiency in scenarios with: (i) long propagation delays, (ii) minimum to relatively high packet error-rate, and (iii) intermittent connectivity.

**Keywords:** Delay-Tolerant Networking, Reliable Transport Protocol, Space Communications

## 1 Introduction

The success of space missions primarily relates to fulfilling their communication needs. However, as space missions increase and grow in complexity, there is an accompanying desire for interoperable missions. The Consultative Committee for Space Data Systems (CCSDS) [1] and other research groups active in the space communications area, are investigating ways to shift to such an interoperable communication platform [2]. Ability to exploit different combinations of spacecraft as data relays to Earth can, indeed, offer alternate communication opportunities; increase data return volumes; and reduce mission operating costs.

Delay-Tolerant Networking (DTN) architecture has been proposed as a means to interconnect heterogeneous network regions, which feature long message round-trip

times and/or interruptions in connectivity [3, 4]. DTN essentially conceals the underlying protocol stack and lays the communication standards for data exchange interoperability.

Other research endeavors concentrate on applying standard Internet technologies into space-based communication networks [5, 6]. The Internet paradigm provides a successful example of how heterogeneous networks cooperate and jointly offer services to each other. It is still under investigation whether proper adaptation of the Internet protocol stack is feasible for space communications as well. However, integrating the space-based infrastructure with the terrestrial Internet can seamlessly extend IP services in space, and spread the use of well-known Internet applications on-board spacecraft.

Whichever approach will eventually prevail (whether separate or combined deployment of DTN and IP, or even a different network scheme), an appropriate transport approach should be utilized. Space communication resources are limited, and are not expected to grow substantially in the foreseeable future. Along with the requirement for efficient use of scarce bandwidth resources, another major objective for space communications is reliable data transfer. Reliability is typically obstructed by high error rates present in space communication channels.

In this paper, we present Delay-Tolerant Transport Protocol (DTTP): a reliable transport protocol specialized for space communications. DTTP is important for space environments because it supports in-network storage that strengthens it against intermittent connectivity. Indeed, DTTP operates efficiently in intermittently available network paths where TCP-like transport approaches cannot even operate.

However, network entities (such as spacecraft, base stations on Moon or elsewhere, relay satellites, Earth ground stations, etc.) are the building blocks for internetworking into Space. On the basis of scheduled or predicted connectivity, and even during opportunistic contacts, stateful sessions take place. In such stateful connections, a DTTP agent is signaled as to when it can communicate to a peer node, and available information (such as the capacity of the communication link and the duration of the connection) is utilized by DTTP's rate-based transmission mechanism. Given that aggregate data return volumes for space-to-space or space-to-earth connections are limited, DTTP's ability to efficiently utilize individual links (as they become available) coupled with its parallel data transfer functionality, underline DTTP's importance, performance-wise.

In addition, DTTP's transmission behavior acts in an asynchronous manner, in contrast to standard TCP's strict compliance to closed-loop communication rules. The reason being that a space entity (at least in current predetermined space connections) solely uses a link (or a percentage of its capacity) for a certain period of time. Thus, DTTP acquires available bandwidth resources via its rate-based transmission behavior, and only adapts its sending rate upon receipt of explicit notification that congestion is present (possibly in the form of storage resources depletion). As a result, DTTP proves robust against packet error rates that can reach relatively high levels. In essence, DTTP primarily targets reliability, efficiency, and interoperability among space missions. We focus on space communications, though applicability of DTTP in other delay-tolerant network environments might be possible as well.

The remainder of the paper is organized as in the following. Section 2 presents work related to transport protocols for space communications. In Section 3, we de-

scribe DTTP features, potential network architecture, and implementation issues. Experimental evaluation of DTTP is presented in Section 4. Finally, Section 5 concludes the paper.

## 2 Related Work

TCP, the Internet's Transmission Control Protocol, is widely recognized to perform poorly across geostationary satellite links of around 0.5sec round-trip time [7]. This is due to the TCP's exponentially-growing probing of available capacity during slow-start, and its understanding of fairness as it interprets every packet loss as being a sign of congestion and thus slows its sending rate. TCP is shown to be less suited to larger delays due to the interaction of various timers present in TCP implementations [8]. TCP could be used in a direct Earth/Moon communication, though it could not effectively use the available link capacity. In greater distances and especially in scenarios characterized by intermittent connectivity, TCP essentially cannot function at all. In the following, we briefly review some transport approaches designed to function in space environments.

TP-Planet [9] is a reliable transport protocol designed for space environments that tries to capture the available link resources. It presumes the presence of IP infrastructure in space, and deploys a rate-based additive-increase multiplicative decrease congestion control. Yet, TP-Planet does not fully exploit available channel capacity because it *emulates* Slow Start and Congestion Avoidance algorithms of conventional TCP. TP-Planet is not tested over a multihop scenario but pertains to a single deep-space link. Also, it relies on conversational functions in order to set data rates. Thus, it implicitly expects continued bidirectional connectivity between the sender and the destination (which is not applicable in deep space communications).

SCPS-TP [10, 11] is the proposal of the Consultative Committee for Space Data Systems (CCSDS) for space communications. It consists a transport protocol based on TCP with appropriate extensions for space environments. SCPS-TP uses header compression and Selective Negative Acknowledgement (SNACK) options, to address limited bandwidth and provide more efficient loss recovery. When there is indication of congestion, either standard TCP or TCP-Vegas congestion control mechanisms can be used to provide congestion control. Otherwise, SCPS-TP operates in open loop rate control mode, where the transmission rate of outbound traffic is limited on the basis of a fixed rate parameter or on the basis of feedback from remote systems. In other words, SCPS-TP is based on existing TCP protocols with some modifications and extensions, which are shown to be inadequate for addressing the challenges in interplanetary networks (for example, TCP-Vegas cannot fully utilize space links due to its window-based nature and the apparent difficulty to accurately measure the variation in RTT for such long distances).

CFDP [12] is a protocol suitable for transmission of files to and from spacecraft data storage. In essence, it operates by copying files from a source storage medium to a target storage medium. Both unreliable and reliable (based on Negative Acknowledgements) services can be offered. The *core* functionality is the simplest form of operation and pertains to file delivery across a single link. The *extended* mode of opera-

tion refers to more complicated scenarios, and provides store-and-forward functionality across a network containing multiple links with disparate availability.

Authors of [13] propose LTP-T to target deep space applications. LTP-T uses the notion of *custody transfer*: each LTP-T entity must accept custody for all blocks it successfully receives. Custody is thus passed from host to host until the final destination is reached. This can minimize end-to-end data delivery time, though the requirement of accepting custody for all received blocks can also lead to storage exhaustion problems. The protocol supports delay-tolerant transport and congestion notification. However, LTP-T is only defined in a generic way, and further details of how it would operate in reality are not provided in the paper.

### 3 Delay-Tolerant Transport Protocol

#### 3.1 Features

Delay-Tolerant Transport Protocol (DTTP) provides reliable data transfer through challenged network environments. It is a transport protocol suitable for network environments characterized by intermittent connectivity, high bit-error rates, and long propagation delays. DTTP comprises a packet-oriented transfer approach that is designed to accelerate data transfers. Indeed, a fundamental design principle for DTTP is to improve data transfers' speed via efficient use of (possibly limited) communication resources.

Basic features of DTTP are summarized as follows:

(i) *Reliability*.

Though DTTP can operate in unreliable mode upon demand, ensuring and accelerating reliability is one of the core design goals for DTTP. Reliable transfer is assured by acknowledgments sent by the receiver. However, unlike TCP's ACK-clocking algorithm, DTTP exploits ACK information in an asynchronous manner: sending and acknowledgment processes in DTTP are *loosely* coupled when compared to standard TCP's behavior. In particular, acknowledgment procedures do not confine the sending rate, which is set autonomously based on bandwidth availability, presence or absence of storage capacity, congestion, and other network conditions. This property makes DTTP suitable for intermittently connected environments, such as space communications.

(ii) *Custody transfer*.

DTTP adopts the notion of custody transfer, upon which, the Delay-Tolerant Network architecture [3] is built. In essence, reliable transfer responsibility is delegated from original sender to next available intermediate node across the communication path, and this process can further be repeated until the final destination receives all data sent. Splitting the communication path into a distinct set of consecutive, reliable connections can serve a number of purposes; most notably, it alleviates senders from having to buffer large amount of data. Once data is received by next node on the path and custody is accepted, that data can be deleted from sender's buffer even if it has not reached its final destination. Consider, for example, the case of a robot on Mars with limited storage and energy resources that is able to dispense its data load to a

base station in range; or a rover on the Moon that distributes its data to a satellite crossing by. DTTP comprises a packet-oriented transport approach. Thus, in order to maximize efficiency, a packet that is accepted by next available custodian and which has yet to be acknowledged to the originating peer, can be immediately forwarded to the next-hop. This approach minimizes the time required for complete end-to-end delivery of the data.

(iii) *Parallel data transfer.*

Space agencies and industrial associates worldwide are involved in structuring standard technologies to achieve interoperable missions. In the years to come, space communications are expected to shift from current statically-organized communication sessions toward a more flexible network architecture. To exploit this upcoming networking framework, DTTP is equipped with the option of parallel data transfer. Thus data transfer can be accomplished in parallel data paths, exploiting various communication opportunities. Sequence of application data is resumed at the receiver. Parallel data transfer can be implemented by preserving the original sequence number space. For instance, assume a rover on the Moon is about to transmit a file (partitioned in 10,000 packets) to a ground station on Earth, and decides to accomplish this via two separate orbiting satellites. The rover splits the data payload, and forwards one portion (sequence numbers 0 through 4,999) to one satellite, and the remaining data (sequence numbers 5,000 through 9,999) to the second satellite, when a contact opportunity appears. This feature requires explicit definition in the protocol header, so that the final destination can anticipate and merge data packets coming from different paths.

(iv) *(Time periods with) constant sending rate.*

DTTP is a rate-based protocol. It employs a constant sending rate in order to exploit the available bandwidth at its full capacity. During scheduled (or even opportunistic) contacts, the protocol steadily fills the pipe. A primary design goal of DTTP is thus to efficiently exploit the valuable space communication resources. As explained next, sending rate can be adjusted according to network conditions.

(v) *Sending rate adaptivity.*

DTTP's sending rate can be accurately characterized as *temporarily constant*. Sending rate adaptation can follow network events such as storage capacity exhaustion. These network events can be either *perceived* by senders through advanced mechanisms or explicitly signaled by receivers; however, in this paper, we do not elaborate on relevant network conditions and how they should affect sending rate.

(vi) *Application-oriented transmission behavior.*

Depending on application type, DTTP can select (or better be instructed to use) a suitable transmission strategy. When no application hint is given, DTTP shows a default transmission behavior. However, we consider that various application types can benefit from customized transmission strategies. The notion of adaptive transport behavior is further explained in Section 3.3, where we present two implementation examples.

### 3.2 Potential Architecture

There is a lot of research effort concentrated on how interoperability between different space agencies can be achieved. Different network protocols, various link layer technologies, static versus dynamic routing, are some of the conflicting approaches regarding space internetworking.

Probably, one of the hot arguments is related with the suitability of IP in space. Indeed, IP extension into space might constitute the common network functionality to interconnect space networks, and integrate them with the terrestrial Internet. However, there exist open issues regarding IP adaptation to space environments. Assuming an IP-everywhere communication infrastructure, DTTP in such a networking framework is shown in Fig. 1.

Recently, a lot of attention has been brought around Delay-Tolerant Network Architecture. DTN's major characteristic is that it can glue together different network protocols. More specifically, DTN acts at the application layer and essentially connects *network regions* that potentially run different protocol stacks. Under this scenario, DTTP's custody transfer functionality is deactivated since it is also offered by the DTN architecture. Figure 2 shows DTTP in a DTN-enabled internetwork.

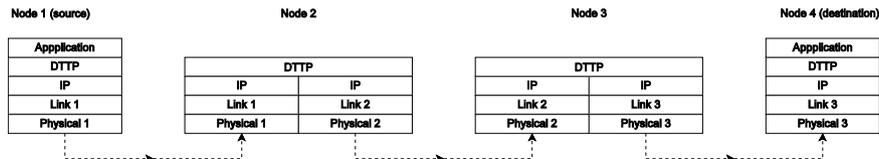


Fig. 1. DTTP deployed in an IP-enabled internetwork.

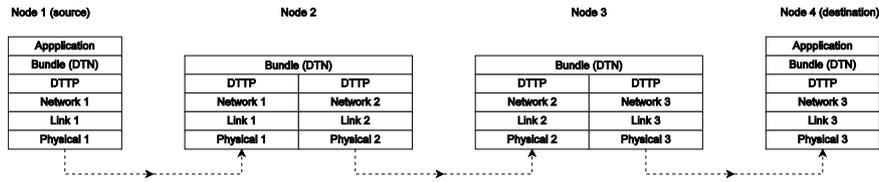


Fig. 2. DTTP deployed in a DTN-enabled internetwork.

### 3.3 Implementation

We have implemented DTTP's primary functionality in Network Simulator ns-2 [14]. Our first implementation of DTTP is based on SACKs. However, we intend to evaluate various acknowledgment mechanisms, such as selective acknowledgments (SACKs), selective negative acknowledgments (SNACKs), delayed acknowledgments, or come up with new sophisticated acknowledgment functions.

In accord with different application scenarios, we present two general tactics to provide reliability. The first targets applications that can benefit from *graduated reliability enhancements*. Suppose, for example, the case where a 30-minute video cap-

tured by a distant satellite is sent back to Earth. Also assume it is desired that each minute is reliably received for instant play-back previously to full-video receipt. An analogous case is a high-resolution photography that can be received in steps, each one providing a better resolution of the photo. Relevant DTTP implementation requires almost immediate use of acknowledgment information as it arrives, and possibly sending *redundant* data (e.g. doubly transmitting the same data segments in order to avoid or minimize data loss due to bit errors of the wireless channel). Note that this approach might waste some portion of network capacity, due to retransmissions and proactive duplicate-transmissions of data segments. A simplified pseudo-code for this transmission approach is the following:

```

until (all application data is acknowledged)
  start transmitting new application data
  if (acknowledgment info arrives)
    send or multiply-send missing data
  end;
end;

```

The second tactic targets bulk-data transfers, which form a very common application type for stressed networks. Indeed, speed of light limit coupled with rare/short communication opportunities restrict other application types (e.g., real-time applications) from functioning properly, especially when propagation delays become extremely long. A suitable algorithm is to: (i) send all application data, from first to last segment; (ii) exploit available acknowledgment information and retransmit missing segments; and (iii) iterate the receive-ACKs and retransmit processes until done. This transmission strategy is described in the following instruction set:

```

send all application data
until (all application data is acknowledged)
  exploit current acknowledgment info
  send or multiply-send missing data
end;

```

Both transmission tactics (and how receipt of packets is affected) are depicted in Figures 3 and 4. Of course, in reality, the number of received packets is orders of magnitude higher. The figures present how gaps of missing packets (at the receiver) are filled.

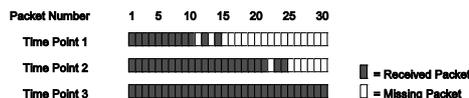


Fig. 3. First transmission tactic.

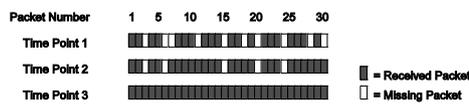


Fig. 4. Second transmission tactic.

## 4 Experimental Evaluation

### 4.1 Evaluation Methodology

We have implemented DTTP’s primary capabilities on the Network Simulator [14]. In particular, our DTTP implementation supports reliability (based on the second transmission tactic described in Section 3.3, and deploying SACK functionality), custody transfer, and rate-based transmission up to the line capacity. We have left parallel data transfer capability as a future extension to the protocol. However, given the static nature of space communications, we have also omitted congestion control. Indeed, each communication session in current space missions involves extensive planning and prior scheduling. Should space communications shift to more flexible architectural schemes, congestion control can certainly be incorporated into DTTP.

In this preliminary study, we evaluate DTTP performance over space network paths varying in distance, data rate, number of hops, and error rate. In our simulations, distance extends through to Mars-Earth communication scenarios. In such distances, data rates are in the order of hundreds of Kbps, such as in ESA’s Mars Express mission and NASA’s Mars Reconnaissance Orbiter. The maximum bandwidth-propagation delay product (which reflects the size of the communication pipe) occurs at shorter distances where high data-rates dominate that product [15]. At this stage, a *comparative* experimental evaluation is not feasible. On the one hand, TCP is not a competitive candidate for comparison due to its inability to function over very long distances with intermittent connectivity, and on the other hand, we do not have sufficiently detailed description to produce reliable simulation code for protocols such as LTP-T.

Table 1 shows the configuration of simulation parameters: combinations of Round Trip Times (RTTs) and data rates considered in our simulation tests; number of hops; and packet error rates. Error models that more precisely capture space network conditions were not explored in our study, and are left for future work. Indeed, wireless channels typically exhibit burst error characteristics. Relevant to distances covered in our tests, space communication channels demonstrate bit error rates of  $10^{-5}$  through  $10^{-8}$ . We just apply a uniform packet error rate that spans up to 10%, and examine the effects on DTTP performance.

**Table 1.** Simulation parameters.

Parameter	Value
{Round Trip Time (sec), Data Rate (Mbps)}	{500, 10} {1200, 2} {2500, 0.5}
Number of Hops	2, 5
Packet Error Rate (%)	0, 1, 5, 10

In our tests, we use DTTP on top of IP, and packets are 1000Bytes long. In every simulation there is one DTTP source (the original sender) that wants to transfer a 10MByte file (i.e., there is no contention for channel capacity). Two topologies are used: a 2-hop and a 5-hop topology, as depicted in Fig. 5. For each direction (downlink or uplink), the one-way propagation delay is uniformly distributed among

all links comprising the path. The packet error rate is equal for both the forward and return path (that carries the application data and the acknowledgments respectively). Along with applying the afore-mentioned error-rate, last link in the forward path exhibits intermittent connectivity: it constantly alternates between *on/off states* (i.e., connected/disconnected) at a fine-grained granularity relevant to each simulation’s duration. On average, that last link is available for 70% of the simulation time, and unavailable for the remaining 30% of time. Also, for this preliminary set of tests, buffer capacity of each DTTP entity is set to relatively high levels (a few MBytes suffice for our tests) so that no data or ACK packet is lost due to a buffer being full. Instead, packet losses are caused only by link errors introduced in the simulations. However, our intention in this paper is to validate DTTP’s properties and provide some initial results for a file transfer scenario.

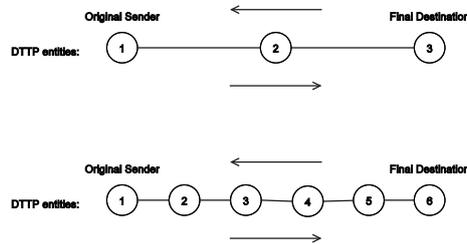


Fig. 5. 2-hop and 5-hop topologies.

## 4.2 Results and Discussion

Graphs in Fig. 6 present completion times for a 10MByte file transfer, both in 2-hop and 5-hop topologies. Each of the graphs refers to a combination of RTT and channel capacity values, as laid out in Table 1.

First noticeable observation is DTTP’s ability to grab available bandwidth. Indeed, in all cases DTTP completes the file transfer in about twice the value of one-way propagation delay. An interesting (and expected) result is that DTTP performs better when the number of hops increases. This is a rational outcome that is justified by the custody transfer capability of DTTP. More specifically, as reliability task is delegated from one node to the next one, shorter SACK-feedback receipt times (now pertaining to distinct links and not to end-to-end paths) facilitate data communication and bring shorter application completion times. Closer look at the graphs in Fig. 6 reveals that as error rate increases, then the greater the number of hops the better DTTP performs. This is depicted by the deviation angle of 2-hop and 5-hop lines in Fig. 6.

We also remark that varying degrees of error rate does only slightly affect the file delivery completion time, when a DTTP session is given full provision of the communication channel. This is explained by the fact that a single DTTP application runs on all links in our simulations. Therefore, DTTP exploits bandwidth resources at the maximum extent: if all application data has been sent at least once, then the (original or intermediate) DTTP sender *fills the pipe* by multiply transmitting missing segments. Of course, this is reflected in the percentage of retransmitted packets (not presented here due to space limitations). The trade-off between file delivery completion

time and percentage of retransmitted packets affects the end-system and overall network performance, and is to be investigated in a future study of ours. We refer the reader to [16] for an automatic retransmission technique that induces early packet retransmissions in order to cope with high bit error rates.

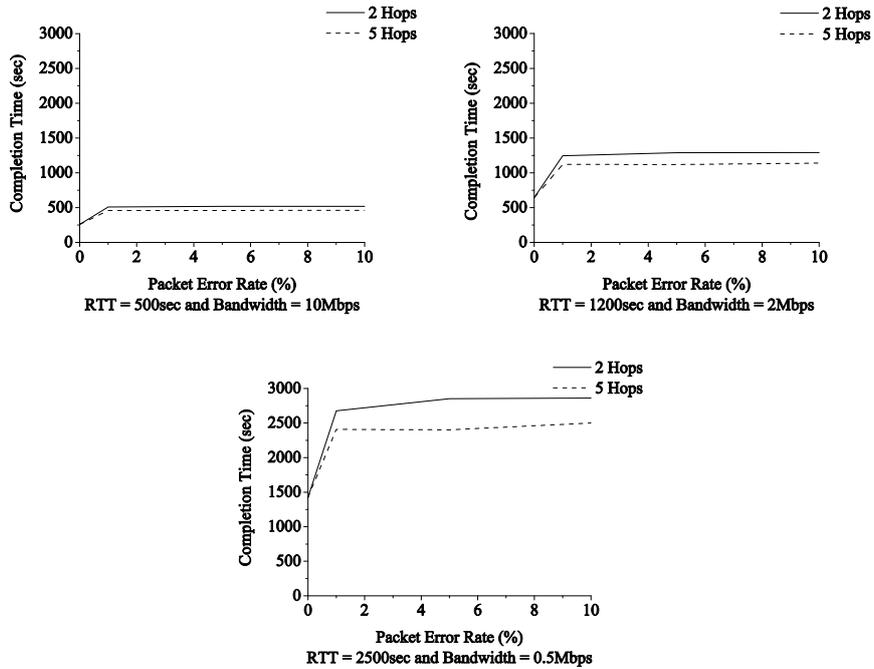


Fig. 6. File delivery completion time using different communication pipes.

The impact of propagation delay on DTTP throughput is shown in Fig. 7. Again, we observe that the greater the number of hops, the better the DTTP throughput achieved. In particular, when the (average) packet error rate increases, DTTP benefits more drastically from greater number of network hops.

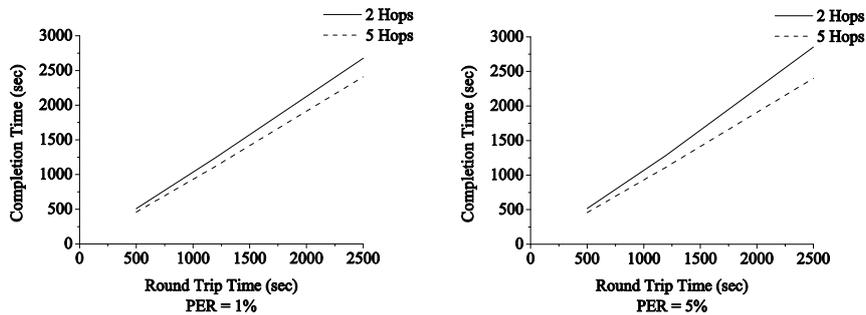


Fig. 7. Round-trip-time impact on file delivery completion time.

## 5 Conclusions

Current and planned activities in space allow for communications strategies that rely on collaborative and multihop end-to-end paths. In this context, a reliable transport service is clearly an important missing component of the communication architecture. We proposed and evaluated DTTP as a potential candidate to fill this gap. We discussed how DTTP satisfies the functionality requirements of both DTN- and IP-oriented space communications. We also demonstrated experimentally that DTTP exhibits satisfying performance, which is enhanced as number of hops increases. In essence, DTTP forms a reliable, transport solution to support flexible multi-node space communications.

As a future task, we are planning to refine and add to DTTP functionality, and validate its properties further. More specifically, we intend to: (i) explore efficient acknowledgment schemes in order to cope with extreme bandwidth asymmetries; (ii) improve its (re-)transmission behavior taking into account the limited bandwidth resources and possibly in relation to delay-bandwidth product; and (iii) supplement DTTP with new capabilities, such as data transfer via parallel paths and explicit signaling for storage resources exhaustion.

## References

1. The Consultative Committee for Space Data Systems (CCSDS), The Official Web Site, <http://www.ccsds.org/>.
2. Ian F. Akyildiz, Ozgur B. Akan, Chao Chen, Jian Fang, and Weilian Su, "InterPlaNetary Internet: state-of-the-art and research challenges," *Computer Networks*, Vol. 43, Issue 2, pp. 75-112, October 2003.
3. V. Cerf et al., "Delay-Tolerant Networking Architecture," RFC 4838, Internet Engineering Task Force, April 2007.
4. K. Scott, S. Burleigh, "Bundle Protocol Specification," RFC 5050, Internet Engineering Task Force, November 2007.
5. K. Hogue, E. Crisculo and R. Parise, "Using standard Internet Protocols and applications in space," *Computer Networks*, vol. 47 no. 5, pp. 603-650, April 2005.
6. Operating Missions as Nodes on the Internet (OMNI), <http://ipinspace.gsfc.nasa.gov/>.
7. C. Partridge and T. Shepard, "TCP Performance Over Satellite Links," *IEEE Network*, vol. 11 no. 5, September/October 1997.
8. Lloyd Wood, Cathryn Peoples, Gerard Parr, Bryan Scotney and Adrian Moore, "TCP's protocol radius: the distance where timers prevent communication," *Third International Workshop on Satellite and Space Communications (IWSSC '07)*, September 2007.
9. I.F. Akyildiz, O.B. Akan and J. Fang, "TP-Planet: A Reliable Transport Protocol for Inter-PlaNetary Internet," *Journal on Selected Areas in Communications*, Vol. 22, Issue 2, pp. 348-361, February 2004.
10. R. C. Durst, G. J. Miller, and E. J. Travis, "TCP extensions for space communications," in *Proceedings of MOBICOM '96*, November 1996.
11. Consultative Committee for Space Data Systems, "Space Communications Protocol Specification-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.*

12. Consultative Committee for Space Data Systems, "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.
13. Stephen Farrell, Vinny Cahill, "LTP-T: A Generic Delay Tolerant Transport Protocol," Technical Report TCD-CS-2005-69, Trinity College, Dublin, 2005.
14. The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>.
15. Jay L. Gao, John S. SeGuí, "Performance Evaluation of the CCSDS File Delivery Protocol – Latency and Storage Requirement," IEEE Aerospace Conference, March 2005.
16. Ioannis Psaras, Giorgos Papastergiou, Vassilis Tsaoussidis and Nestor Peccia, "DS-TP: Deep-Space Transport Protocol," IEEE Aerospace Conference, March 2008.