# Where Does Transport Layer Fit into Space DTN Architecture?

Giorgos Papastergiou, Christos V. Samaras and Vassilis Tsaoussidis
Dept. of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi, Greece
{gpapaste, csamaras, vtsaousi}@ee.duth.gr

*Abstract*— **Delay Tolerant Networking (DTN) has been proposed as a global overlay architecture to provide network connectivity in challenged environments such as deep-space communications field. Bundle Protocol along with its convergence layer protocols transform end-to-end reliability into questionable hop-by-hop reliable communication and imposes a two-tier routing policy that allows flexible routes only within regions. Following the OSI model, in this paper we discuss the advantages and disadvantages of having a transport protocol relying on DTN for providing reliable and transparent transfer of data between end systems and offering common transport layer services such as end-to-end packet-oriented retransmission, flow control, path diversity and redundant transmission. In that context, we highlight features from DS-TP and DTTP that correspond to the requirements of transport protocols for space. We present space communication scenarios, which cannot be adequately addressed by the current DTN architecture.**

***Keywords: Delay Tolerant Networking; deep-space communications; transport layer.***

## I. INTRODUCTION

Space communications typically rely on scheduled links and require less sophistication from communication protocols. In this context, the link layer was the dominant layer for space communications; routing was never an issue; end-to-end reliability was frequently overlapping with reliability of a single hop; congestion and overflow were absent due to strict scheduling of communication activities and admission control; and the limited required sophistication was shifted to the application layer.

However, two new major properties have changed the spectrum of potential architectural choices for space communications: (i) the multihop architecture, which is required to reach deep space and (ii) the increasing number of alternative communicating paths that may be used to reach a single receiver. Inline with these two properties, the demand for interoperability among space agencies has also contributed towards the emerging field of *Delay Tolerant Networking* (DTN) [1]. The DTN architecture [2] and the accompanying Bundle Protocol (BP) [3] specification documents present a means for data communication on potentially heterogeneous networks characterized by high delays or disruptions. However, the DTN architecture alone cannot cope with these challenged environments. It essentially glues together dissimilar protocol stacks, but it relies upon underlying network services adapted to the special networking conditions.

DTN's storing functionality and its property to delegate custody seem to match two major requirements for space communications: one is the requirement for permanent storage due to long interruptions in space communications; another is the requirement for hop-by-hop reliable communication until the receiving end.

We depart our discussion from the somewhat misleading name of "Delay-tolerant networking": Delay-tolerant networking does not mean necessarily a delayed service or a service where minimal delay is not important. Instead, delay-tolerant service means a service that can be provided even if the network imposes delays or disruptions. However, under circumstances of disruptions or delays, the service may - and perhaps should - be relatively fast.

Therefore, we emphasize here the potential of a "store and Forward" service instead of a "Store and forward" service, which appears, falsely in our opinion, as the default service of DTN networks. In other words, DTN efficiency in multihop networks can be promoted when emphasis is placed on flexible and prompt forwarding procedures for (even partially received) DTN messages. In this context, we investigate here the potential role of a transport layer within the DTN scheme; where does such a layer fit; what functions should it carry; and when it may be necessary.

A transport layer cannot be naturally replaced by application-level functions; and cannot be replaced by link layer functions. At the application layer, the potential advantage of packet-oriented feedback for scheduling the retransmissions is lost and we necessarily deal with messages. A transport layer can assist end-to-end reliability, bandwidth exploitation for the whole end-to-end path and, in the future, congestion control as well. The transport layer handles reliability on a per packet basis, whenever feedback is gathered from the receiver.

Although DTN architecture depends on regional transport protocols, such as Licklider Transmission Protocol (LTP), TCP or UDP, in this paper we claim that enhanced sophistication can be incorporated into an end-to-end transport layer above BP. This sophistication pertains to reliability, dynamic routing and multiple path exploitation, congestion control, and also the

978-1-4244-6833-1/10/$26.00 ©2010 IEEE 

exploitation of the delay/bandwidth tradeoff for the benefit of faster and reliable services.

We depart our discussion from the observation that hop-by-hop reliability cannot guarantee application reliability. This is not a new argument; rather, it has attracted major attention from the internet community and resulted in the well-known "end-to-end argument", which claimed that end-to-end monitoring allows for more complete administrative approaches than the administration of intermediate hops. We exploit the occasional necessity of an end-to-end transport service with a case study, which we call "the missing bundle". In the context of our scenario, we show that the Bundle Protocol does not really provide an end-to-end service, as it is falsely claimed in the RFC 5050.

Furthermore, end-to-end service, which is typically implemented by the transport layer, allows for optimal route selection when source routing is deployed. Source routing can exploit any alternative path to destination, any time, unlike hop-by-hop routing, which may result in path loss due to connectivity disruptions. We uncover the confines of simultaneous data forwarding and custody transfer which increases the risk for data isolation in space and discuss the benefits of having additionally the source as the responsible entity until data delivery has been confirmed by the receiver. This approach requires a new design, with multiple custodians. Along the same lines, we also highlight a distinctive requirement of DTN routing which should become a source-oriented service, unlike the typical IP routing.

In addition, we discuss the possibility of end-to-end congestion control at the transport layer and we contrast it with the comparative hop-by-hop congestion control. Although congestion control in space is not an issue yet, it will become inevitably an issue as soon as DTN is deployed along with dynamic routing. So far, the DTN architecture has not yet incorporated solutions for that problem.

Finally, we exploit the potential of transport protocols to trade bandwidth for delays, using redundant information. We present the details of Deep-Space Transport Protocol which adjusts its proactive reliability service to bandwidth availability and network conditions. We present results that confirm the efficiency of such scheme.

Although end-to-end communication promises a wide spectrum of useful functions, it is often tightly-coupled with closed-loop systems. That is, transmission control is dominated by the use of receiver feedback (i.e., through acks) and, hence, becomes occasionally inefficient (for example, in deep space communications). We highlight later in the paper the possibility to distinguish end-to-end transmission control from end-to-end functionality that allows for feedback, which however, does not regulate data traffic.

The structure of the paper is organized as follows: in Section II we briefly discuss recent transport layer approaches for space communications. In Section III we identify the limitations of current space DTN architecture and investigate the service gap that an end-to-end (transport layer) service can fill. Finally, in Section IV, we conclude on the basic requirements for a transport layer protocol that could possibly enhance future space communications.

## II. RELATED WORK

Prior to the adoption of DTN architecture as the future technology for space internetworking, a number of proposals regarding transport layer networking over deep-space links have been introduced. One of the early proposals for reliable data transmission over deep space links is TP-Planet [4]. 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.

A similar approach, which comes from the same authors, is the unreliable RCP-Planet [5] protocol. RCP-Planet incorporates a probing rate control scheme to cope with link congestion and error rate, in conjunction with a packet-level Forward Error Correction (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.

Space Communications Protocol Standards - Transport Protocol (SCPS-TP) [6] 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 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.

CCSDS File Delivery Protocol [8] is an Application Layer protocol, but it also provides Transport Layer functionalities, such as detecting and retransmitting corrupted or lost data. Therefore, it can run over an unreliable network protocol such as the Space Packet Protocol [9] and UDP. 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.

Even though CFDP has been designed to deal with the space-ground communication constraints, it appears to be unsuitable when communication complexity increases (e.g., dynamic routing or file distribution through multiple relay points in parallel). Therefore, a solution that is currently under investigation is the integration CFDP with the Bundle Protocol, where each CFDP PDU is encapsulated in a "bundle" (a BP datagram). Under such setting: CFDP PDUs are independently routed in store-and-forward fashion through the network; CFDP is active only at the endpoint entities; and the function of point-to-point retransmission devolves to end-to-end acknowledgement.

In the context of Delay Tolerant Networking, Bundle Protocol (BP) [3] has been proposed by the Delay Tolerant Networking Research Group (DTNRG) [10] as the overlay protocol that implements the DTN architecture [2]. The main features of BP:

• *Store-and-Forward Message Switching:* The bundle protocol deploys store-and-forward message switching. The protocol data unit of the bundle protocol is the so-called *bundle*. More specifically, the bundle layer transforms each application data unit into one or more bundles.

• *Permanent In-Network Storage:* Permanent-storage devices provide data store for possibly long periods.

• *Custody Transfer:* Custody transfer allows the source to delegate retransmission responsibility to the next available node on the path to the destination.

• *Reliability:* Reliability is facilitated by the custody transfer feature. Nevertheless, only coarse-grained retransmission is optionally deployed. Reliable data delivery service is assumed to be provided by the underlying protocols.

• *Naming and Late Binding:* DTN nodes are identified by (at least) one text-string identifier, which is expressed syntactically as a Uniform Resource Identifier (URI). Late binding allows name-based routing, while destination name-to-address translation need not be performed at the source node.

• *Routing:* DTN provides routing across potentially heterogeneous networks. Routing computation algorithms remain a challenging task under investigation.

• *Fragmentation:* A large application data unit or bundle may be fragmented in order to utilize the contact periods more efficiently.

In the same context, Licklider Transmission Protocol (LTP) [11] has been proposed as a DTN convergence layer for deep-space links. 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.

In [12] we have introduced Deep-Space Transport Protocol (DS-TP) as a transport layer scheme of choice for the space networking protocol stack. 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 on the a priori-known and predetermined line-rate. This way, DS-TP achieves high link-utilization from the beginning 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.

Delay-Tolerant Transport Protocol (DTTP) supports reliable data transfers in delay-tolerant networks [13]. We have designed DTTP as a stand-alone transport protocol for multihop space network topologies. To combat network partitions and extreme delays, DTTP employs store-and-forward model for packet switching, that is coupled with in-network persistent memory. Furthermore, DTTP can operate over multiple paths in parallel, and incorporates dynamic routing functionality: each DTTP node can autonomously and dynamically (that is, based on network performance measurements) select how to route incoming DTTP packets towards their ultimate destination. We have performed relevant simulation tests and validated DTTP performance advantage against protocols that rely on static routing procedures. We underline that, though DTTP features have been developed to support space communications, DTTP can apply to other challenged networks as well.

Lately, erasure coding has attracted some attention in the context of data transmission over challenged networks. Sophisticated FEC-based techniques such as FLUTE [14] have the potential to improve the system's performance; see for example [15], 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. 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).

Finally, the authors of [16] propose the addition of a session layer to the DTN service model in order to deal with several shortcomings of the DTN architecture. More specifically, this session layer offers partial ordering among bundles belonging to the same session, along with a publish/subscribe group membership service model, which simplify the construction of several information distribution DTN applications.

## III. DO WE NEED AN END-TO-END TRANSPORT PROTOCOL?

In this section, we investigate the need for end-to-end (transport layer) services above Bundle Protocol, which are currently missing from the DTN architecture. Bundle Protocol forms a common space network layer that provides global naming and routing, custody transfer, and permanent storage. Based on specific scenarios, we investigate whether it is advantageous to deploy a transport protocol above BP. Such a transport protocol can exploit BP's features and will allow for end-to-end reliability, dynamic routing, efficient exploitation of alternative paths, storage congestion control, and variable store-and-forward policies.

### *A. Hop-by-hop reliability*

DTN architecture is defined in RFC 4838 [2] and RFC 5050 [3], and has been adopted by CCSDS to form the future space internetworking architecture [17]. It comprises a common network layer structure to transfer data across possibly heterogeneous links. Bundle Protocol provides the overlay network on top of various terrestrial and space-oriented protocol stack configurations. However, no mechanism or protocol does have a complete view of the network, and thus end-to-end functionality is being solely left to the applications. In that context, DTN employs custody transfer, a hop-by-hop approach to allow for optional reliable data delivery by means of shifting the data reliability responsibility to the upcoming nodes towards the destination. Additionally, the reliable service from the underlying network can enhance reliability further. Licklider Transmission Protocol provides such a service. Although bundle delivery progress can be reliably checkpointed along the path to the destination, the lack of a global view of network resources availability cannot guarantee successful bundle delivery.

For our discussion, we consider the scenario shown in Fig. 1. This scenario includes a Mission Control Center (MOC) on Earth communicating through multiple orbiters with a remote in-situ network on another planet. We assume that all nodes (including the nodes belonging to the remote network) implement BP and support different underlying network technologies. In particular, TCP/IP-based DTN communication is provided between MOC and Ground Stations (GS), as well as among the nodes of the remote planet surface network, while CCSDS-based DTN communication (TM/TC/AOS/Proximity-1) [18] is supported on the space links. Fig. 2 shows a possible protocol stack configuration. We further assume a predetermined routing scheme that is based on the prior knowledge of relevant movement of the space elements. Each node makes routing decisions based on link connectivity schedules [19]. This scenario constitutes a representative example of future long-term space communications. Based on this scenario, we describe a case study where hop-by-hop reliability cannot guarantee successful delivery.

When a bundle destined to a node on the remote surface network is generated at the MOC, a routing decision regarding the next hop is made and the bundle is forwarded to GS-1 with the custody transfer option enabled. GS-1 either accepts or denies custody, calculates next hop of the path based on link connectivity schedules with respect to the bundle's lifetime, and then forwards the bundle. We focus our discussion on the case where that bundle is actually transmitted to Orbiter-1 for forwarding to the remote surface network, Orbiter-1 accepts custody and for some reason (e.g., unpredicted disruption, erroneous contact schedule information or node failure) the data cannot be transmitted during the intended contact. At that moment only Orbiter-1 retains a copy of that bundle, since custody acceptance signal received by GS-1 removes any bundle's retention constraints at the GS-1.

We divide our discussion into two sub-cases:

Case 1: We assume that a failure at Orbiter-1 has caused the deletion of the bundle from its local storage and has rendered orbiter non-operational. Even if MOC may have requested for a bundle deletion report, in the best-case scenario this report will be transmitted only when Orbiter-1 becomes operational again. If failure lasts for a long time, this feedback information received by MOC through a GS will probably be useless. In the worst-case scenario that the failure is fatal, no bundle deletion notification report will be ever generated. Although this case scenario may not be very common in future space communications, transferring the custody on intermediate nodes in the path will always include that threat. Based on this infrastructure, application-level mitigation services are needed to deal with this "missing bundle problem", such as timer expiration and data retransmission. However, these services match better the inborn characteristics of a transport layer.

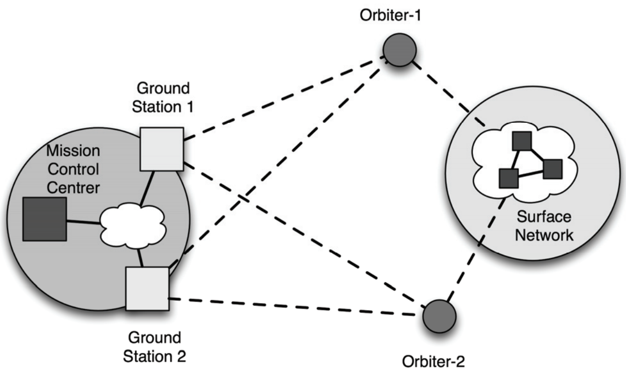


Figure 1. A typical future space communication scenario.

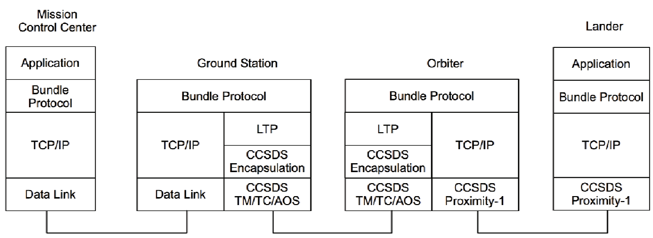


Figure 2. Protocol stack configuration.

A sophisticated transport protocol running end-to-end could detect missing bundles prior to failure recovery and retransmit the bundles through an alternative path, if any (e.g., through Orbiter-2). Such a transport layer protocol may incorporate advanced mechanisms (such as transmission suspension and timer-based retransmissions) that can efficiently exploit the available contacts schedule information, and can possibly be enhanced by Bundle Protocol's hop-by-hop custody transfer and reporting mechanisms.

Case 2: Unlike the previous example, here, we investigate reliability further in the context of timely delivery; that is, in case of unpredicted disruptions or erroneous routing decisions, in what extent can BP exploit any alternative communication opportunities to deliver a bundle prior to the expiration of its lifetime? Although this condition is clearly correlated with the routing scheme employed by BP or the underlying network infrastructure, it highlights the need for an end-to-end mechanism that exploits alternative communication paths. We consider the same scenario described above with the only difference being that an unexpected disruption occurs during the intended Orbiter-1 - Lander contact, rather than a complete failure. BP retains a copy of the bundle and will either choose to wait until connectivity is restored or to re-route the bundle via GS-1/GS-2 and Orbiter-2, trying to deliver the bundle prior to its lifetime expiration. If both approaches fail then the bundle is discarded from the network, even though an alternative path from MOC to Lander may exist. Placing a sophisticated transport protocol on top of BP that retransmits proactively bundles based on path observations will lead to successful delivery of "stranded" bundles.

Following the above discussion, we conclude that reliability in space communications imposes two requirements; a) the ability to detect and handle missing data, and b) the timely delivery of data with respect to their useful timeline. We have showed that in case of unexpected service disruptions or node failures, BP cannot assure application reliability due to the lack of an end-to-end functionality. End-to-end mechanisms are still needed in order to ensure reliability; and such mechanisms can be offered by sophisticated transport layer protocols placed architecturally above BP.

### B. Path exploitation

Bundle Protocol hides the underlying protocol stacks upon which it operates, and requires routing information that connects DTN nodes at the application layer. That does not preclude routing protocols at a lower layer. Instead, routing can actually occur hierarchically at two levels: inside each autonomous system (assuming a set of heterogeneous networks), and also between the higher-level DTN nodes. For example, the planetary surface network shown in Fig. 1 may constitute an IP-based autonomous network. Some nodes of this group may serve as DTN gateways between local and deep-space communications. Bundle Protocol terminates local networking protocols at these gateway nodes and passes data to the Orbiters using a different underlying infrastructure.

Given the prior knowledge pertaining to the relevant movement of planets, moons, and spacecraft (namely, landers and relay satellites), a more-or-less static/deterministic routing system is expected to govern space internetworking. A deterministic routing framework can at least characterize the space network backbone, which constitutes ground stations (either on Earth, Moon, Mars, or elsewhere) and relay satellites with predetermined orbits. Bundle Protocol can offer such capability (see Contact Graph Routing [19] for example).

Although scheduled contacts correspond to the requirement for predetermined paths, lack of an end-to-end service limits the capability of an application to exploit efficiently path availability. This situation gets worse when BP's custody transfer option is enabled. As the responsibility for retransmission propagates towards the final destination, hop-by-hop-based routing makes network more vulnerable to failures due to unexpected disruptions. In Section III.A, we have presented a scenario where an unexpected contact disruption caused the disposal of a bundle, although an alternative end-to-end path was available. This scenario revealed the need for an end-to-end transport layer service on top of BP that allows for source-based routing when hop-by-hop approach fails.

Another issue that dictates further the need for source-driven routing decisions is optimal route selection. In the future, sophisticated DTN-oriented routing scheme may be developed and incorporated into BP, which will account for several path characteristics such as contacts schedules, storage availability, and Packet Error Rate (PER). Such routing schemes will make routing decisions based on these path characteristics to choose an optimal path regarding reliable and timely delivery of bundles. A transport protocol, in this case, could offer an end-to-end monitoring service that based on some receiver feedback could dynamically check the optimality of the routing decision and optionally dictates the re-forwarding of the data stream through an alternative path when necessary. When BP's custody transfer option is enabled, such a transport protocol would require the retention of bundles at the source node.

Finally, a transport protocol placed architecturally above a network layer, can provide multipath parallel transfer services to enhance reliability and assure faster delivery. In this context, multiple identical copies of a bundle may be transmitted in parallel following different paths towards the destination. This approach trades bandwidth for delays, transmitting redundant information. Sophisticated transport layer protocols can be used to dynamically deal with this tradeoff, based on estimated network conditions. As an alternative to redundant transmission, erasure coding techniques can also be exploited to increase performance further whenever additional processing overhead does not dominate communication delays. An erasure code can take as an input a message of $k$ bundles and encode it into an $n$-bundles message (where $n > k$) for a code rate $r=k/n$. In the ideal case the original message can be recovered from any subset of $k$ bundles.

### C. Storage Congestion Control

An essential element of delay-tolerant networks is that bundles need to wait in queues for potentially long timescales due to long delays and disruptions. Smooth operation of store-and-forward message switching relies upon two assumptions:

(i) that memory resources are well-distributed throughout the network, and (ii) that storage is persistent to combat network disconnections and long/variable propagation delays.

Storage becomes a valuable resource in DTNs that needs to be carefully protected and managed. In essence, network congestion takes the form of storage congestion: depletion of storage resources at some node (i.e., the receiving node) causes it to discard all incoming bundles. To make matters worse, sending nodes cannot release memory resources occupied by stored bundles, since the bundles get discarded at the receiving node. As a result, storage congestion propagates backwards affecting other nodes as well.

Data route selections in delay-tolerant networks need to take memory resources availability and probability of congestion into account. Storage congestion control in DTNs is an ongoing research issue. Bundle protocol can only locally respond to congestion. More specifically, bundle protocol response to memory resources exhaustion may be to: delete already-stored bundles whose lifespan is reached; discard incoming bundles; or accept incoming bundles but refuse custody for them. Nevertheless, bundle protocol lacks knowledge of network resources availability and network connections dynamics, that render it incapable of preventing or alleviating storage congestion.

On the other hand, transport layer mechanisms can complement the DTN architecture and control congestion events. One transport layer solution is to apply flow control on a hop-by-hop basis. The sending node can dynamically adapt its transmission rate so that it does not exceed the average rate that the receiving node can accommodate (that is, without dropping received bundles).

A second solution to storage congestion is to adapt data path in accordance with congestion level. End-to-end transport services allow for more sophisticated approaches on congestion control. In particular, information about network connectivity schedule enables an end-system to select an alternative data path for its bundles. So, without the need of manual and global network administration, end-systems can actively contribute to: network congestion control; better in-network memory availability; network and application performance enhancements.

### *D. Forwarding Policy*

Bundle Protocol bases its operation around the special notion of its data unit, the so-called *bundle*. A bundle can relate to application data (or data coming from the layer above the bundle layer) in various ways, depending on the network environment and present application types. Thus, a bundle may: (i) contain a single file (e.g. a photograph), multiple files (such as small-sized engineering telemetry files), or a file segment (possibly part of high-quality video); (ii) be of fixed size determined by application type or network management procedures; (iii) be of variable size set by application, and containing a coherent bundling of application data, e.g. a set of data records that can be independently processed; (iv) be self-contained and include self-described metadata useful for the application at the receiving end-system. Bundle Protocol specification does not limit bundle size or define content of bundles. One design principle highlighted in the specification is the use of variable-length bundles (possibly large in size) instead of data streams or limited-sized packets, with a view to potentially enhance network ability for good path selection decisions.

We underline that memory resources at DTN senders (forming either edge or intermediate network nodes) are freed on a bundle basis. That is, a bundle is removed at once after its custody is accepted on the receiving node and relevant acknowledgement is propagated back to the sending node. On error-prone links exhibiting extremely long delays and disruptions (such as space links), relatively large bundles might throttle network performance. More specifically, partial transfer of bundles (either due to limited connection time or link errors) may noticeably increase in-network memory requirements, since large portion of bundles are in essence replicated and stored on both receiving and sending sides of network links. On the other hand, use of small-size bundles may increase protocol overhead (thus wasting valuable bandwidth resources) or cancel valuable properties of bundle contents (such as, self-contained messages or self-explanatory application chunks that can be independently processed at the receiving application).

In addition, Bundle Protocol specification largely amplifies store procedures for bundles, and degrades offered forwarding services. In particular, a receiving node awaits receipt of each bundle in its entirety before it further forwards it towards its final destination. As a consequence, successfully transmitted bundle parts are detained on DTN receivers, engage valuable memory resources, and data flow is at least partly confined.

We argue that custody transfer functionality can and should be decoupled from forwarding policy. Delay-tolerant networking does not necessarily translate into belated communications. Minimizing data delivery time comprises a key goal by its own right, even in presence of extreme delays and network partitions. To that end, proposed DTN architecture lacks certain functionality and needs supportive mechanisms of other protocols in the network protocol stack. For instance, DS-TP protocol promotes data delivery via its dynamically adjusted double retransmission mechanism: a relatively small percentage of bandwidth is utilized to proactively prevent or amend packet errors. Also, DTTP protocol advances bundle delivery and reduces in-network memory requirements through dynamic and parallel path selection based on network performance measurements. DTN architecture can be further supported by similar mechanisms in order to: accelerate bundle delivery, even if bundle fragmentation is to be applied; provide flexible routes based on bundle size, data priority, network performance measurements, or other criteria; better administer memory resources and thus enhance data production rates and space missions efficiency.

### *E. Trading bandwidth for delay using redundant data transmissions*

As already stated in Section III.B, an end-to-end (transport layer) protocol incorporating redundant transmission mechanisms can significantly enhance BP services, in terms of reliability and delivery latency, by exploiting available parallel paths. In this section, we elaborate on this issue in more detail.

Proactive redundant transmissions impose by definition a trade-off between efficient bandwidth exploitation and faster delivery. Indeed, reduction in data delivery time can be achieved through trading some bandwidth for delay. This holds true especially for space network settings, where long propagation delays and intermittent connectivity are common. However, given that space communication resources are scarce and expensive, efficient exploitation of available resources is necessitated. The following question then naturally arises: in what extent does extra and redundant transmission effort enhance data delivery without wasting bandwidth resources?

This optimal point of operation corresponds to an optimal redundant transmission rate. The retransmission rate is a function of (i) application constraints (such as delivery timeline and data priority), and (ii) path characteristics, such as: available bandwidth; propagation delay; packet error rate (PER); and communication contacts duration.

In [12] we introduced DS-TP, a transport protocol that includes a proactive retransmission scheduling policy, the so-called Double Automatic Retransmission (DAR). DAR allows for efficient and fast retransmission of corrupted data packets. Our mechanism proactively transmits redundant packets and adjusts its retransmission rate based on the experienced packet error rate. DAR can be easily extended to account for other parameters as well.

Preliminary results reveal that using some percentage of bandwidth to proactively prevent packet corruptions does not always assure faster delivery. We show a representative graph that depicts this behavior. These simulation results pertain to a file transmission between two nodes using the DAR mechanism. Link bandwidth, round-trip time (RTT), and file size equal to 1Mbps, 300 seconds, and 4xBandwidth-Delay Product, respectively. Fig. 3 presents file transfer time with respect to the applied link PER. DAR retransmission rate ranges from 0 to 0.5. We note that a retransmission rate equal to 0.5 essentially means a double file transmission.

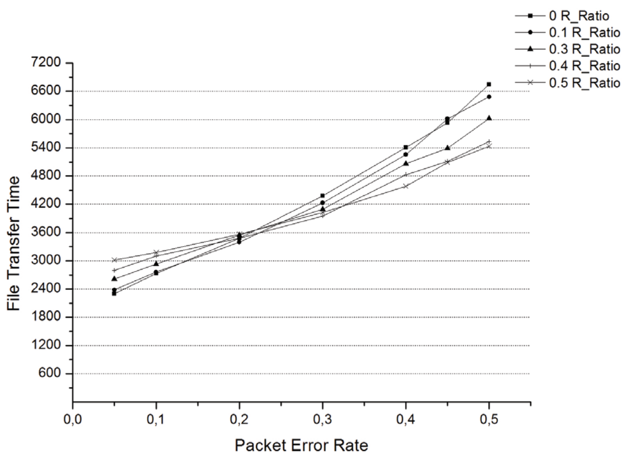


Figure 3. File Transfer Time using redundant transmissions.

We observe that, for small packet error rates, file transfer time is minimized when no redundant data transmission is utilized at all. 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. However, as packet error rate increases, DS-TP and its DAR mechanism show clear advantage and complete file delivery faster.

Although DS-TP has been introduced as a transport layer scheme to enhance hop-by-hop reliability in space DTN, DAR mechanism can be incorporated into an end-to-end transport protocol as well. In this case, redundant transmissions can propagate through multiple parallel paths towards destination. This will make redundant transmission mechanisms more flexible, but introduce more complexity at the same time.

Finally, we note that using packet-level erasure coding instead of pure redundant transmissions will allow for advanced recovering from packet losses. Coding rate can be dynamically adjusted based on network measurements. Several coding schemes can be applied, such as Reed-Solomon and Low Density Parity Check (LDPC). Although using packet error correction decreases retransmission overhead for the same packet recovering capability, it essentially increases processing overhead. That is, there is a tradeoff between processing overhead, retransmission overhead, and recovering capability.

## IV. CONCLUSIONS

When delay is an issue for space, but also when application reliability is required within a predetermined timeframe, DTN architecture in its current form may not suffice. In this discussion we concluded that hop-by-hop reliability does not necessarily guarantee application reliability; simple custody transfer may not allow for timely completion of application task; hop-by hop routing may not exploit the best available path from source to destination; and the lack of end-to-end service may not permit optimal decisions for congestion control.

We have highlighted occasional benefits from a true end-to-end transport protocol for DTN, and summarized the requirements of such protocol, such as the decoupling of custody transfer and data forwarding service, the exploitation of Delay/Bandwidth tradeoffs for the benefit of proactive reliability and the possibility to incorporate multiple paths in the forwarding service. Our intention is to implement and evaluate this service extensively in the near future, using ION DTN reference implementation and the ESA/ESOC DTN Testbed, which we host in our Lab.

## REFERENCES

[1] S. Burleigh et al., "Delay-Tolerant Networking – An Approach to Interplanetary Internet", IEEE Communications Magazine, July 2003.

[2] V. Cerf et al., "Delay-Tolerant Network Architecture", IETF RFC 4838, information, April 2007, http://www.ietf.org/rfc/rfc4838.txt.

[3] K. Scott and S. Burleigh, "Bundle Protocol Specification", IETF RFC 5050, experimental, November 2007.

[4] 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.

[5] J. Fang and I.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.

[6] "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.

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

[8] "CCSDS File Delivery Protocol (CFDP) Part 1: Introduction and Overview", Informational Report, CCSDS 720.1-G-3. Green Book. Issue 3. April 2007.

[9] "Space Packet Protocol". Recommendation for Space Data System Standards, CCSDS 133.0-B-1. Blue Book. Issue 1. September 2003.

[10] Delay Tolerant Networking Research Group. http://www.dtnrg.org/.

[11] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Specification", IETF RFC 5326, experimental, September 2008, URL: http://www.ietf.org/rfc/rfc5326.txt.

[12] G. Papastergiou, I. Psaras, and V. Tsaoussidis, "Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs", Special Issue of Elsevier Computer Communications on Delay/Disruption- Tolerant Networking, Vol. 32, Issue 16, pp. 1757-1767, October 2009.

[13] Christos V. Samaras and Vassilis Tsaoussidis, "Design of Delay-Tolerant Transport Protocol (DTTP) and its Evaluation for Mars", Acta Astronautica, Elsevier, accepted for publication.

[14] T. Paila et al., "FLUTE: File Delivery over Unidirectional Transport", RFC 3926, October 2004.

[15] D. Kutscher, J. Greifenberg and K. Loos, "Scalable DTN Distribution over Uni-Directional Liks", in Proceedings of NSDR'07, August 27 2007, Kyoto,Japan.

[16] M. Demmer and K. Fall, "The design and implementation of a session layer for delay-tolerant networks", Computer Communications, Vol. 32, Issue 16, pp. 1724-1730, October 2009.

[17] "Rationale, Scenarios, and Requirements for DTN in Space". Draft Informational Report, CCSDS. Draft Green Book. March 2010.

[18] "Space Data Link Protocols – Summary of Concept and Rationale". Informational Report, CCSDS 130.2-G-1. Green Book. December 2007.

[19] S. Burleigh, "Dynamic Routing for Delay-Tolerant Networking in Space Flight Operations", in Proceedings of SpaceOps'08, May 2008, Heidelberg, Germany.