The Journal of Systems and Software 84 (2011) 226–237

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Adjusting transport segmentation policy of DTN Bundle Protocol under synergy with lower layers

Christos V. Samaras*, Vassilis Tsaoussidis

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

ARTICLE INFO

Article history:
Received 10 July 2010
Received in revised form
16 September 2010
Accepted 18 September 2010
Available online 29 September 2010

Keywords:
Delay-Tolerant Network Architecture
Interplanetary Networking
Space Internetworking

ABSTRACT

We assess Delay-Tolerant Network (DTN) performance in space under the scope of adjusting protocol data unit (PDU) size at various layers. We quantify the importance of combinatively adjusting size of DTN bundles, transport packets, and link frames. Through simulations, our paper reveals trade-offs that involve file delivery time, transmission effort of sending nodes, and memory resources release rate. Based on our findings, we propose a transport adaptation scheme that dynamically adjusts DTN bundle and transport packet size by means of heuristic search. To our knowledge, this is the first study to examine transport segmentation policy and interaction among various layers of the DTN protocol stack.

1. Introduction

Until recently, it has been common practice among space missions to (i) operate each spacecraft in isolation of others via direct-to-Earth communication links, (ii) manually schedule, review and revise communication contacts between all spacecraft and ground station antennas on Earth, (iii) inspect incoming data as it arrives and send commands to spacecraft in order to request retransmission of data either missing or received in error, and (iv) often deploy customized and incompatible protocol stacks tailored to accommodate each mission's specific needs. This anachronistic model of operation poses significant limitations on total communication time provided to spacecraft requires tedious and error-prone human intervention and, essentially, forces mission designers to "reinvent the wheel" every time an ad hoc communication system and protocol stack needs to be deployed on some spacecraft. Likewise, space communications rely heavily on proprietary protocols and confidential products, which often leads to incompatible communication protocols among different agencies. In turn, this limitation does not allow for exploiting space resources for the benefit of science, efficiently. Therefore, some form of automated space internetworking architecture appears necessary.

In this context, Delay-Tolerant Networking (DTN) appeared as an emergent solution to the problems mentioned earlier and quickly gained wide acceptance in the space community. DTN architecture forms a message-oriented overlay that offers message relay services across potentially long propagation delays or even network disruptions (Cerf et al., 2007). It employs persistent storage to withstand network interruptions, and includes a hop-by-hop transfer of reliable delivery responsibility. DTN Protocol Data Units (PDU) are called *bundles*, and may carry application data along with information needed to deliver them to final destination. Key DTN functionality is that each bundle is kept in memory in its entirety, and is deleted upon receipt of acknowledgment for its successful delivery to next node on the path to destination. However, DTN Bundle Protocol specification (Scott and Burleigh, 2007) does not limit bundle size or specify content of bundles. Indeed, 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. Given the way DTN bundles are released from memory and the reliance on persistent memory, memory storage becomes a critical resource on Delay-Tolerant Networks that calls for careful management.

Here, we address the issue of proactive and adaptive fragmentation policies that can be employed by Delay-Tolerant Networking. In particular, we study the consequences of various packet and bundle sizes on network and buffer management performance, aiming at a conclusive strategy for scheduling transport services within space internetworking activities. Our study mainly exploits the

* Corresponding author. Tel.: +30 2541079553; fax: +30 2541079554.
E-mail addresses: csamaras@ee.duth.gr (C.V. Samaras), vtsaousi@ee.duth.gr (V. Tsaoussidis).

0164-1212/$ – see front matter
doi:10.1016/j.jss.2010.09.023

trade-offs of various fragment selections, occasionally highlighting the significant cost of overhead, the potential gain of throughput or, further, the possibility to exploit memory resources efficiently.

For example, users of Delay-Tolerant Networks and services may establish specific expectations pertaining to enhancing data delivery time, reducing sender transmission effort, mitigating energy resources requirements, and/or releasing buffer resources of participating nodes faster. Depending on the Delay-Tolerant Network protocol stack in question, PDU size for protocols at the various layers may be (i) fixed, (ii) variable and dependent on application type or other factors, (iii) dynamically adjusted, or (iv) yet to be defined/optimized for protocols whose specification either allows variable PDU (and protocol header) size or is currently under standardization process.

Furthermore, given the scarcity of power, memory, and processing resources at space nodes, we examine PDU size configuration at the DTN bundle, transport, and link layers, and especially correlate those settings to improve efficiency of space communications. We quantify the synergy of the afore-mentioned network layers, and demonstrate that data transport segmentation policy can largely affect a number of network performance drivers, such as overhead payload trade-off, buffer release capability, retransmission granularity, and routing flexibility. To our knowledge, this is the first attempt to investigate DTN bundle size impact on network and application performance, under the scope of synergy with PDU sizes at lower layers. While our study concentrates on space networks, our findings may be of importance to other DTN environments as well, and our approach can certainly be utilized to other network settings. Beyond that, our study can also be applied to space internetworking environments that use network overlays, other than DTN, to bind heterogeneous network protocols.

The rest of this paper is organized as follows. In Section 2 we discuss the related work and provide some necessary background. In Section 3 we develop a case study to evaluate the impact of transport segmentation policy on DTN performance. In Section 4 we elaborate on our experimental methodology and evaluation plan. We present our results in Section 5 and, based on our findings, we propose a generic approach for packet size determination in DTNs in Section 6. Finally, we conclude the paper in Section 7.

2. Related work

Lately, significant progress has been made in the development of space data handling standards under the auspices of Consultative Committee for Space Data Systems (CCSDS). All major space agencies and space-related scientific and industrial entities collaborate and contribute in the standardization procedures, which are driven by key goals such as enhancement of interoperability and cross-support, and reduction in risk, development time and project costs. Currently, more than 400 space missions use CCSDS-recommended protocols.

Well-established and commonly deployed CCSDS standards cover data link layer protocols; radio frequency and modulation systems; data archiving; data transfer services that move space link data units between ground stations, control centers and end-user facilities; time code formats; and data compression. Standardization of network and transport layer protocols has not reached consensus yet. However, a popular network architecture that is gaining ground and could provide Internet-like services across interplanetary distances is the so-called *Delay-Tolerant Networking architecture* (Cerf et al., 2007).

Space exploration enters a new era characterized by interoperable space missions, automation in reliable data transfers, and networking of space and terrestrial communication assets. In this framework, a series of communication paradigms have emerged over the last years to formulate a modern space network architecture. In more detail, most space missions fly with the CCSDS TM (CCSDS, 2003a), TC (CCSDS, 2003b), AOS (CCSDS, 2006a), or Proximity-1 (CCSDS, 2006b) protocols at the data link layer. Various transport layer solutions have been proposed to support reliable transmission. CCSDS File Delivery Protocol (CFDP) (CCSDS, 2007) forms a hybrid protocol that provides application and transport layer services, is suitable for the transmission of files to and from spacecraft data storage, and can operate across a single link or multiple links via store-and-forward procedures. Licklider Transmission Protocol (LTP) (Ramadas et al., 2008) supports reliable transmissions over extremely long propagation delays, and can serve as the convergence layer below DTN Bundle Protocol and above the data link protocol. Our proposal for challenged networks is the Delay-Tolerant Transport Protocol (DTTP) (Samaras and Tsaoussidis, 2010). DTTP supports reliable data transmissions in space or other networks described by extreme delays and disconnections, runs over multi-hop paths, and incorporates dynamic path selection based on performance probing procedures.

In the literature, studies on space communication performance approach the problem from different standpoints. CFDP storage requirements and file delivery time have been studied in Gao and SeGui (2004) and Lee and Baek (2004). Evaluation of CFDP over DTNs is provided in Koutsogiannis et al. (2009). Sung and Jay (2006) showed weather effect on CFDP performance. Wang et al. (2009) performed an experimental evaluation of DTN Bundle Protocol over LTP for cislunar distances. Dynamic routing techniques for space DTNs are presented in Samaras and Tsaoussidis (2010), Burleigh (2008), and Bisio et al. (2008). Other contributions reinforce space communication reliability through proactive retransmission scheduling (Papastergiou et al., 2009), implementation of erasure-coding schemes within the CFDP itself (de Cola et al., 2007), or rate adaptation based on channel quality (Tarello et al., 2006).

The impact of packet size on throughput performance has been extensively explored for low-delay wireless networks (for example, see Modiano, 1999; Yin et al., 2004; Korhonen and Wang, 2005; Wang et al., 2005; Wu et al., 2007). These works generally concentrate on either the link or the transport layer. On the contrary, there exist only a few papers on optimal packet size determination in space environments. Gao and SeGui (2004) concluded that CFDP PDU size only slightly affects storage requirements, while Wang et al. (2006) examined data throughput with regard to MAC packet size over space links of relatively low delay. However, investigation of cross-layer optimal packet size in space DTNs is only partially explored. Relevant works do not consider the impact of extremely long propagation delays or even link disconnections, nor do there exist any studies on how DTN bundle size affects network performance. Our paper demonstrates PDU size interaction among various layers in space DTNs. To our knowledge, this is the first study to examine transport segmentation policy across the whole DTN protocol stack.

3. Case study: Delay-Tolerant Transport Protocol (DTTP)

In order to quantify the impact of transport segmentation policy on DTN performance, we develop a case study based on our DTTP protocol. DTTP is the protocol selected for implementing the fragmentation policy of choice. However, our study is not DTTP-specific; it can certainly encompass analogous protocols, which operate in DTNs, such as CFDP or LTP. The objective of the study is to reveal the benefits of coordinative setting of PDU size at various layers (namely, size of DTN bundles, transport packets, and link frames).

DTTP encompasses all necessary flexibility for designing the required scenarios, aiming at evaluating the impact of packet size on various communication parameters. We detail DTTP functionality in Samaras and Tsaoussidis (2010), where we also evaluate its performance in space networks. In the following, we provide a concise description of the DTTP protocol.

DTTP core features include: (i) reliability, (ii) custody transfer, and (iii) routing path adaptation. We have designed DTTP with a view to support reliable data transfers in space. However, DTTP can support other challenged network environments as well. It is a conversational protocol, that bases its operation on messages exchanged between sending and receiving nodes. Interaction takes place in an asynchronous manner, so that it withstands long propagation delays or even network partitions. Reliability is guaranteed via acknowledgment packets that travel the opposite direction of data flow. Retransmission timers signal when already transmitted DTTP packets are considered lost and thus require to be transmitted again. Operation of those timers relies closely on the communication schedule: they take into account both long delays and link interruptions, so that pending timers are paused during link disconnection accordingly.

Custody transfer functionality essentially delegates reliable transfer responsibility from one network node to the next across the path to destination. Once a DTTP packet is successfully received at an intermediate node and its custody is accepted, the receiving node notifies the sender, so that the latter can delete that DTTP packet from its memory. DTTP employs persistent storage to combat network partitions and augment the store-and-forward process of custody transfer. We underline that custody transfer of DTTP packets serves mainly two reasons: first, lost or corrupted packets are detected and retransmitted faster on a per-link basis rather than end-to-end; and second, smother network storage occupancy is generally achieved, as the task of data store and reliable transmission is distributed among nodes on the path to final receiver.

DTTP allows for routing path adaptation. We have implemented a mechanism that records performance over available paths and dynamically adjusts data routing. This mechanism runs on DTTP senders autonomously (i.e., without help from the network itself). We have selected *data goodput* as the performance metric to drive route adaptation, which is defined as: size of application data successfully delivered to next node during a communication contact divided by the respective transmission time (in Bytes/s). Of course any other performance metric could be utilized, given that network/application priorities may vary. Performance records for all paths are maintained and regularly updated (e.g., once per communication contact or more frequently) locally at each DTTP sender.

Via management procedures, each node gets assigned a *Node ID*, which forms its unique name within a DTTP network. Local information at nodes enables address look-up capabilities: Node ID gets mapped to corresponding address of the underlying communication system, which may be an IP address, a Path ID of CCSDS Space Packet Protocol, a virtual channel number, etc.

Fig. 1 shows DTTP packet format. It consists the *Header* and *Data* segments, which are 28 Bytes and variable length, respectively. As implied by the names, DTTP *Source Node ID*, *Destination Node ID* and *Current Custodian Node ID* identify the DTTP source entity, the DTTP destination entity and the current custodian of a DTTP packet, respectively. Initially, *Current Custodian Node ID* matches *Source Node ID* but can subsequently point to other intermediate DTTP nodes possibly closer to destination node. Each DTTP connection between a specific pair of source and destination nodes is uniquely identified by *Session ID*, which can thus be used to multiplex/demultiplex different DTTP sessions between the same pair of source and destination entities. *Sequence Number* (of the first byte in the packet's data payload) and *Length* (i.e., size of data payload) guide the receiving node to reconstruct application data order. Additionally, DTTP protects against wrapped sequence numbers (*Timestamp*), and supports error-checking (*Checksum*) and packet fragmentation (*Fragment Offset* and *FR* flag bit).

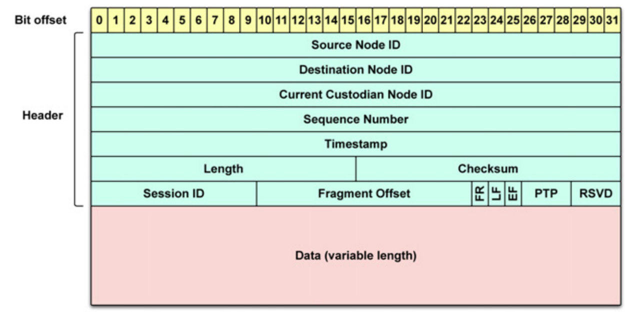

Fig. 1. DTTP packet format.

As explained previously, DTN Bundle Protocol provides message exchange in highly stressed environments. It sits at the application layer over a number of potentially heterogeneous networks. DTTP deployment in the absence or presence of the DTN Bundle Protocol is depicted in Figs. 2 and 3, respectively. In the former case, DTTP exploits in-network permanent storage and memory gets allocated/freed at DTTP packet granularity. In the latter case, DTTP acts as DTN convergence layer, and provides reliable and efficient transport services to the DTN Bundle Protocol: DTTP agents send and receive DTN bundles, rather than interfacing directly with the applications.

4. Experimental methodology

4.1. Simulation environment and configuration

Our evaluation plan has been implemented on the Network Simulator ns-2. In particular:

- We have modified ns-2 simulation platform to support Delay-Tolerant Networks. Typical operation of Internet nodes dictates that, once a link gets disconnected, then the router drops all outgoing packets on that interface. Instead, we have implemented persistent storage on ns-2 network nodes, so that in-network data can withstand link outages and await for a new communication contact.

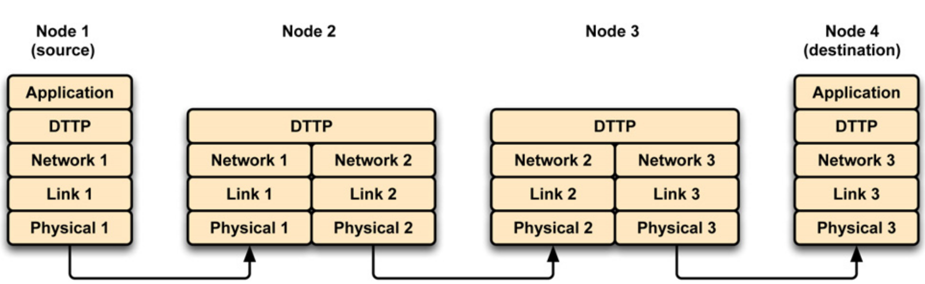

Fig. 2. DTTP deployed in non-DTN-enabled networks.

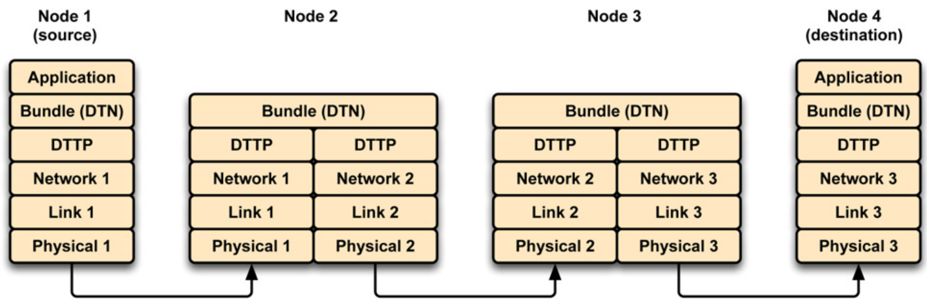

Fig. 3. DTTP deployed in DTN-enabled networks.

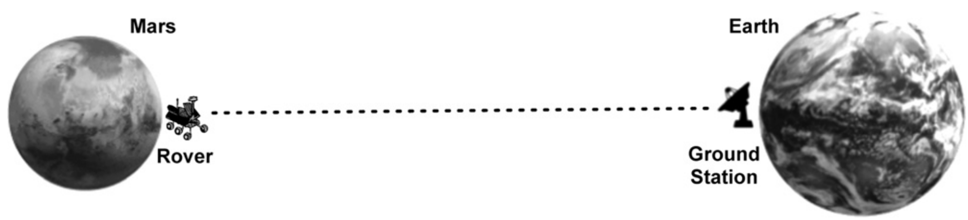

Fig. 4. Network topology.

- We have implemented DTTP on ns-2. As described in Section 3, DTTP enhances reliable data transfers on challenging networks, is delay-tolerant, and provides dynamic multi-path data routing.
- We have implemented a lightweight version of DTN Bundle Protocol (Scott and Burleigh, 2007). More specifically, our Bundle Protocol implementation supports the so-called *bundling process*, that partitions and encapsulates application data inside *bundles*. Most DTN Bundle Protocol features have been omitted, since similar functionality is offered by DTTP.
- We have constructed a reporting module on ns-2 that traces DTN bundles and DTTP packets; records memory utilization on DTN nodes; reports errors; and provides statistics and network measurements at the end of simulations, such as file delivery time, size of retransmissions, etc. Our major modifications and additions to ns-2 architecture and functionality necessitated such a tool.

We consider a deep space scenario where a rover on Mars communicates directly with deep space network antenna(s) on Earth (Fig. 4). Communication schedule is analogous to direct-to-Earth connections available (on the average) to Spirit and Opportunity Mars exploration rovers. As is currently the case, martian rovers can transmit direct to Earth for at most 3 h a day, due to power and thermal limitations, although Earth may be in view much longer. In our simulation set, 2 communication contacts between the rover and an antenna on Earth take place during each martian day (so-called *sol*). Sol duration is 24 h and 40 min, and each communication contact lasts for 1 h. Data transmission rate is set to 12 kbits/s. Propagation delay between Mars and Earth is 10 min, and bit error rate (BER) is assumed to be 10^{-5}, unless otherwise stated. DTN bundle size and DTTP packet size are fixed during each simulation, but they take different values between simulation runs. DTN bundle size ranges from 1 kByte to 1000 kBytes, and DTTP packet size varies from 250 Bytes to 1500 Bytes. DTN bundle header size is 25 Bytes and DTTP packet header size is 28 Bytes. We underline that each DTTP packet is encapsulated in exactly one link layer frame; thus, changing DTTP packet size essentially means adjusting link layer frame size as well (see Fig. 5). Total application data to be transferred amounts to 100 MBytes. Table 1 summarizes simulation parameters.

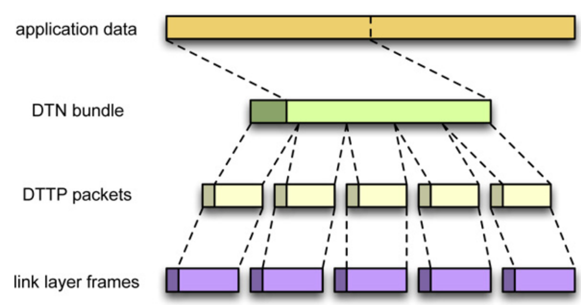

Fig. 5. Protocol stack showing protocol data units (PDUs) at various layers. Dark-colored parts of PDUs denote header segments.

Normally, the delay burden on space data transfers lies on the space segment of the network. Indeed, extraordinary propagation delays and network disconnections in space largely affect file delivery time, whereas the well-connected terrestrial network poses insignificant delays. Therefore, we terminate network topology at ground stations on Earth and omit the ground links between ground stations and Mission Operation Centers.

4.2. Evaluation metrics

In the following, we detail the evaluation metrics utilized in the paper. *File delivery time* is the time when the last byte of application data is received successfully at the receiver. Apparently, it refers to the time when the last missing DTTP packet arrives at destination, which at the same time completes receipt of last missing DTN bundle. We investigate how file delivery time is affected by different

Table 1
Configuration of simulation parameters.

Parameter	Value
Bandwidth	12 kbits/s
Bit error rate	10^{-5}, 10^{-6}, or 10^{-7}
Propagation delay	10 min
File size	100 MB
DTN bundle size	1, 5, 10, 50, 100, 500, or 1000 kBytes
DTTP packet size	250, 500, 750, 1000, 1250, or 1500 Bytes

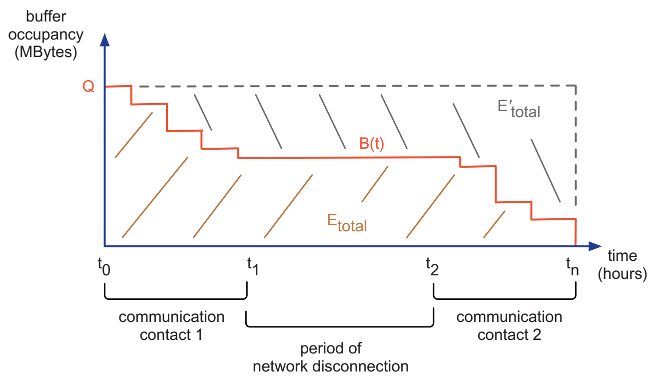

Fig. 6. Definition of Buffer Space Release Rate (BSR) index.

combinations of DTN bundle size and DTTP packet size. Along with total data receipt time, we show *total size of retransmissions* that occur at the DTTP sender till DTTP session termination.

However, depending on PDU size of DTN and DTTP protocols, overhead due to protocols headers can vary significantly at the first place, that is, without taking retransmissions into account. In order to precisely capture additional transmission effort expended during a file transfer, we define *Extra Transmission Effort (XTE)* as total data transmitted by the sender subtracted by application data size (i.e., file size), and normalized by application data size:

$$\mathrm{XTE} = \frac{TotalTransmissionSize - ApplicationData}{Application\,Data}$$

We note that *TotalTransmissionSize* in the above XTE formula pertains to all bytes transmitted at the DTTP level, thus capturing overhead induced by DTN and DTTP protocols plus total retransmission effort. Given that energy resources are often limited in spacecraft, XTE index is valuable for another reason: it essentially reflects *extra energy resources*[1] required for transmitting a file of some specific size. Stated in another way, lower XTE translates into more efficient use of bandwidth resources and results in less energy expenditure.

Delay-Tolerant Networks rely on storage resources available on edge and intermediate nodes. Due to long delays and network interruptions, storage capacity can potentially form the DTN bottleneck, and this calls for careful distribution and administration of memory resources. As explained in Section 1, memory resources get freed up each time a successful bundle delivery is acknowledged back at the sender. Thus, bundle size determines the scale by which memory is resumed every time a bundle is deleted from memory. We graphically present *buffer requirements* of the sender, that depict buffer occupancy at the DTN Bundle Protocol layer.

Faster memory release in DTN networks serves principally a two-fold purpose: (i) spacecraft (e.g., landers, rovers, satellites, etc.) can increase data production rate and thus enhance space mission productivity; and (ii) intermediate DTN nodes (e.g., a relay satellite orbiting a planet) can support flexible data relay services and offer higher multiplexing of data flows originating from multiple sources, as has recently been demonstrated by NASA's Mars Odyssey satellite. In order to quantify how quickly occupied buffer resources are freed throughout a DTN connection, we introduce *Buffer Space Release Rate (BSR)* index (Fig. 6). BSR is expressed as:

$$BSR = 1 - \frac{E_{total}}{E'_{total}}$$

[1] *extra*, in this term, is to be interpreted as protocols overhead and retransmissions.

where

$$E_{total} = \int_{t_0}^{t_n} B(t)\,\mathrm{d}t$$

and

$$E'_{total} = \int_{t_0}^{t_n} Q\,\mathrm{d}t = Q(t_n - t_0)$$

so

$$BSR = 1 - \frac{\int_{t_0}^{t_n} B(t)\mathrm{d}t}{Q(t_n - t_0)}$$

As shown in the figure, $B(t)$ depicts buffer occupancy at the sender; quantity Q refers to initial DTN storage requirements (i.e., application data plus DTN protocol overhead); and time points t_0 and t_n mark overall duration of DTN connection. BSR measures the area E_{total} that corresponds to buffer occupancy during the course of file transfer, and the result is normalized by the area E'_{total} that notionally refers to the hypothetical worst-case scenario (i.e., no memory is ever resumed); then 1 subtracted by the normalized outcome yields BSR, which defines the average rate of memory release. However, DTN connection duration $(t_n - t_0)$, as perceived by DTN sending nodes, varies from one simulation test to the other. Thus, with the intention of normalizing BSR upon a common base, we compute BSR against a constant E'_{total} area: the latter value pertains to the longest DTN connection duration observed in our simulations. As a result, BSR results, presented later in the paper, are directly comparable to each other.

5. Results and discussion

5.1. Data delivery time

In general, intermediate DTTP packet sizes (namely, 500–1000 Bytes) deliver application data to destination faster (Fig. 7). Smaller (250 Bytes) or larger (1250–1500 Bytes) DTTP packets delay file delivery further, and may require additional communication contacts. It is noteworthy that each time an extra communication contact is required for complete file delivery, file receipt time scales up, since two consecutive communication contacts are approximately 12 h apart. That explains all sharp bendings of curves in Fig. 7. We note also that we have come across a few simulation runs, where an additional communication contact was required due to even a single packet missing. The highest point shown on graph in Fig. 7 (namely, simulation scenario exhibiting DTN bundle size of 50 kBytes and DTTP packet size of 1250 Bytes) constitutes such a case. Long delays and network partitions can occasionally cause such inefficient file deliveries.

Table 2 lists time required for file delivery for all simulation runs subtracted by delivery time of best-performing combination of DTN bundle size and DTTP packet size (i.e., case of 50-kByte DTN bundles and 500-Byte DTTP packets). We observe that bundle sizes of about 10–100 kBytes achieve faster file delivery in space. Smaller or

Table 2
File delivery time minus file delivery time of best-performing case in the set (in minutes).

DTTP packet size (Bytes)	DTN bundle size (kBytes)						
	1	5	10	50	100	500	1000
250	760.96	739.99	41.94	59.47	59.15	58.89	58.86
500	42.27	20.11	0.02	0.00	0.02	0.02	0.02
750	50.43	0.04	0.03	0.06	0.05	20.14	20.13
1000	740.00	42.98	0.16	20.23	20.19	20.21	20.21
1250	760.24	740.00	25.25	760.10	45.24	740.00	740.00
1500	761.41	760.41	760.32	58.99	58.72	740.01	760.11

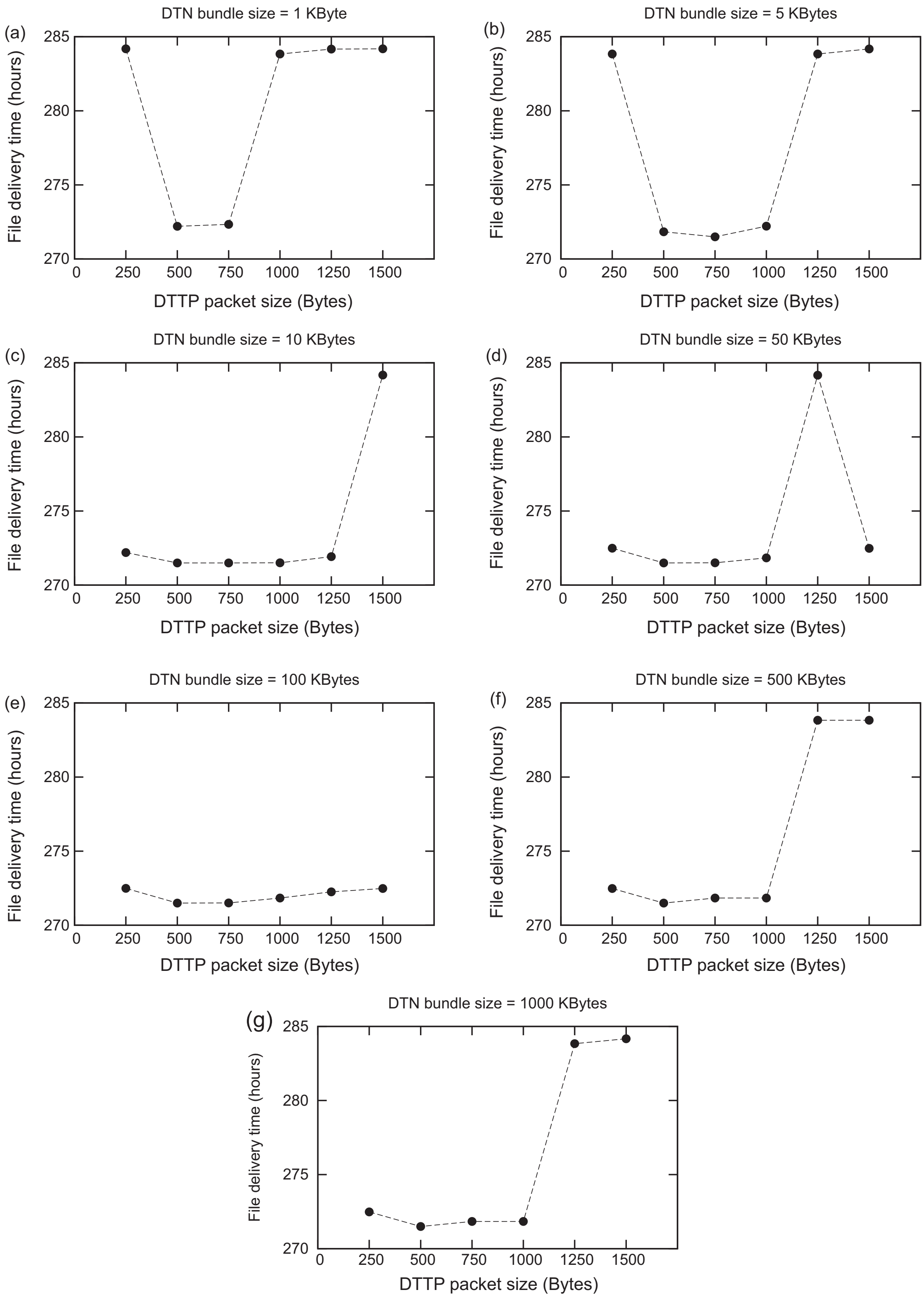

Fig. 7. File delivery time at the destination.

larger bundle sizes delay file delivery by a few minutes up to more than 12 h. The latter cases apparently occur when an extra communication round is required. As shown in the table, decreasing DTN bundle size to 1 kByte causes the greatest file delays. Indeed, very small DTN bundles increase considerably the bandwidth expended for the transmission of DTN protocol headers instead of application data.

5.2. Size of retransmissions and extra transmission effort

In our network architecture, data reliability service is provided by DTTP. Thus, retransmissions of missing data take place at the DTTP layer. Fig. 8 displays DTTP retransmission size at the sending node. For 100 MBytes of application data, retransmissions vary from about 2.3 MBytes to 13.2 MBytes depending on DTN bundle

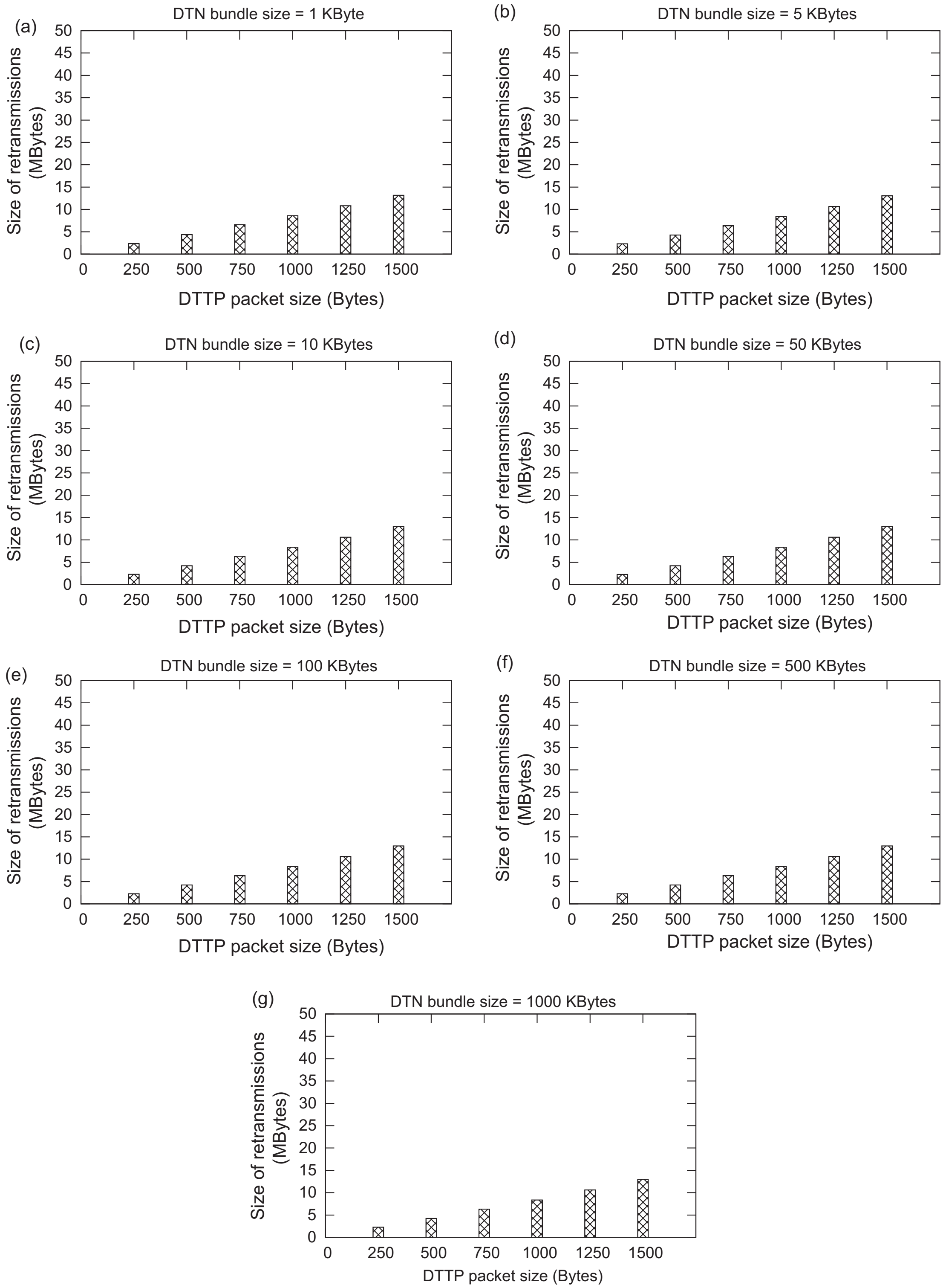

Fig. 8. Total size of retransmissions at the sender.

and DTTP packet size. It is clear from the figure that injecting larger DTTP packets into the network increases overall retransmission size. This observation was somewhat anticipated: given a constant bit error on the space link, the larger the DTTP packet size (and the link layer frame size), the higher the resulting packet error rate.

However, retransmission size by its own right does not reveal the sender's total transmission effort. Total transmission effort refers to transmission of application data, all involved protocols overhead, and data packets retransmissions. For our discussion, we confine protocols overhead to include only DTN and DTTP headers, and exclude overhead introduced in lower layers. For 100 MBytes of application data, total data to be transmitted (without considering retransmissions) ranges from about 102 MBytes (case of 1000-kByte DTN bundles and 1500-Byte DTTP packets) to 115 MBytes (case of 1-kByte DTN bundles and 250-Byte DTTP packets). Extra Transmission Effort (XTE) of the sender is depicted in Fig. 9

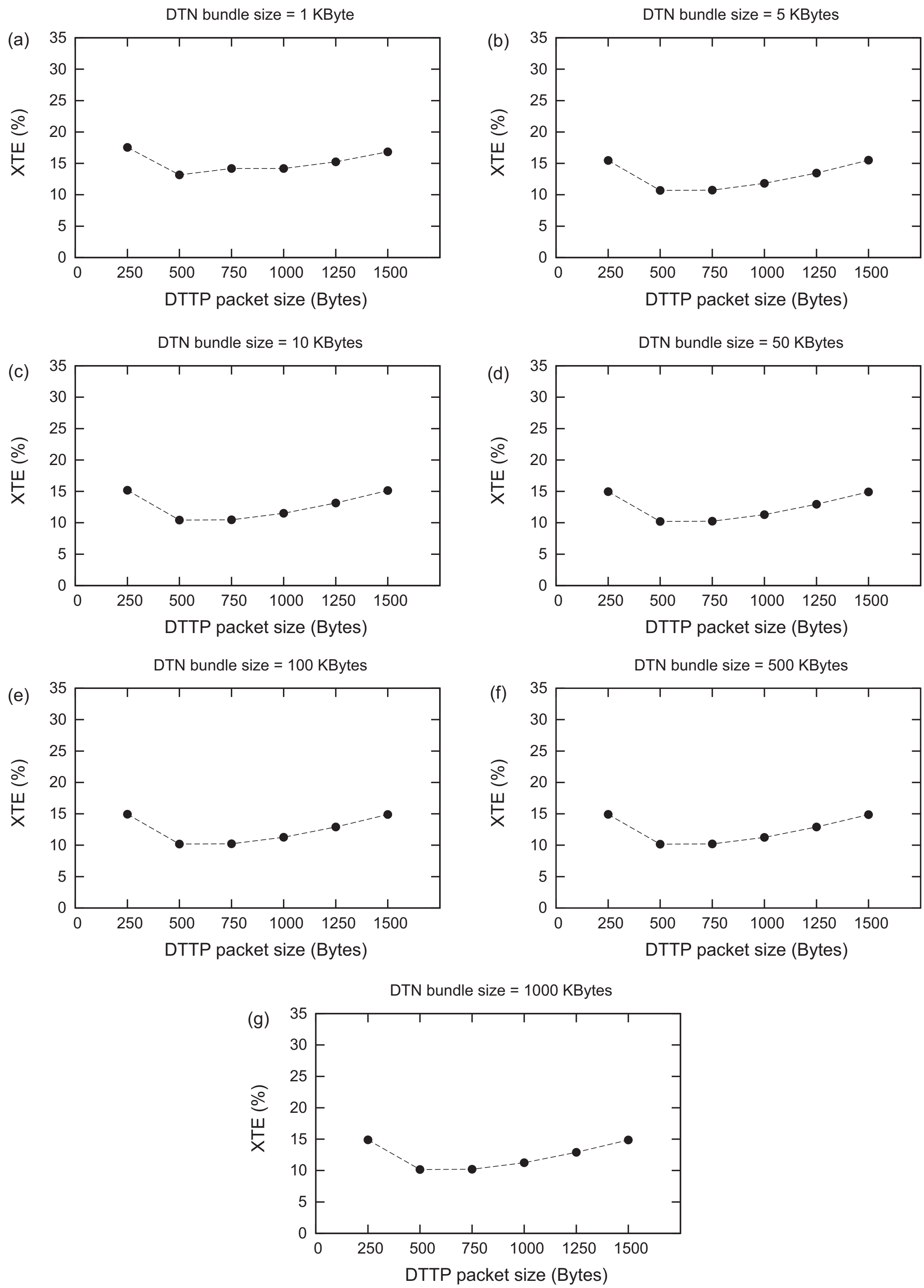

Fig. 9. Extra Transmission Effort (XTE) of the sender.

and shown in Table 3. Noticeably, sender's XTE is mainly determined by DTTP packet size and less affected by DTN bundle size. In particular, DTTP packet size of approximately 500–750 Bytes decrease sender's XTE (and total transmission effort, apparently). On the contrary, DTTP packets larger than 750 Bytes or smaller than 500 Bytes prove inefficient in terms of total transmission effort.

It might have been expected that minimizing DTTP *protocol overhead ratio*[2] by means of utilizing relatively large DTTP packets, would reduce XTE. Nevertheless, when correlating findings of

[2] We define Protocol Overhead Ratio (POR) as the ratio of protocol header size to Protocol Data Unit (PDU) size, e.g. POR_{DTTP} = *DTTP header size*/*DTTP packet size*.

Table 3
Sender's Extra Transmission Effort percent (XTE (%)) for various combinations of DTN bundle size and DTTP packet size.

DTTP packet size (Bytes)	DTN bundle size (kBytes)						
	1	5	10	50	100	500	1000
250	17.543	15.464	15.173	14.954	14.926	14.904	14.902
500	13.161	10.692	10.428	10.217	10.192	10.170	10.168
750	14.178	10.747	10.473	10.257	10.228	10.209	10.204
1000	14.186	11.814	11.502	11.297	11.265	11.246	11.242
1250	15.240	13.461	13.157	12.947	12.911	12.895	12.894
1500	16.837	15.514	15.136	14.913	14.890	14.870	14.870

Figs. 8 and 9, it becomes apparent that larger DTTP packets induce substantially more retransmitted data, and thus eliminate the benefits of lower protocol overhead. Consequently, as DTTP packet reduces in size, sender completes file delivery with less transmission effort. This relation holds consistently for DTTP packet sizes as small as 500 Bytes. For even smaller DTTP packets (e.g., 250 Bytes), extra transmission effort tends to grow: though sizes of retransmissions are minimal, DTTP protocol overhead ratio increases considerably (11.2%) and causes longer file delivery times. The interpretation is that considerable amount of sender transmission effort is expended on transmitting packet headers against application data.

As stated previously and seen in Table 3, DTN bundle size slightly impacts total transmission effort. XTE decreases as DTN bundle size increases. Nevertheless, we observe that DTN protocol impact on XTE weakens for bundle size larger than or equal to 50 kBytes. For those bundle sizes, DTN Protocol Overhead Ratio degrades substantially and thus finer PDU granularity at

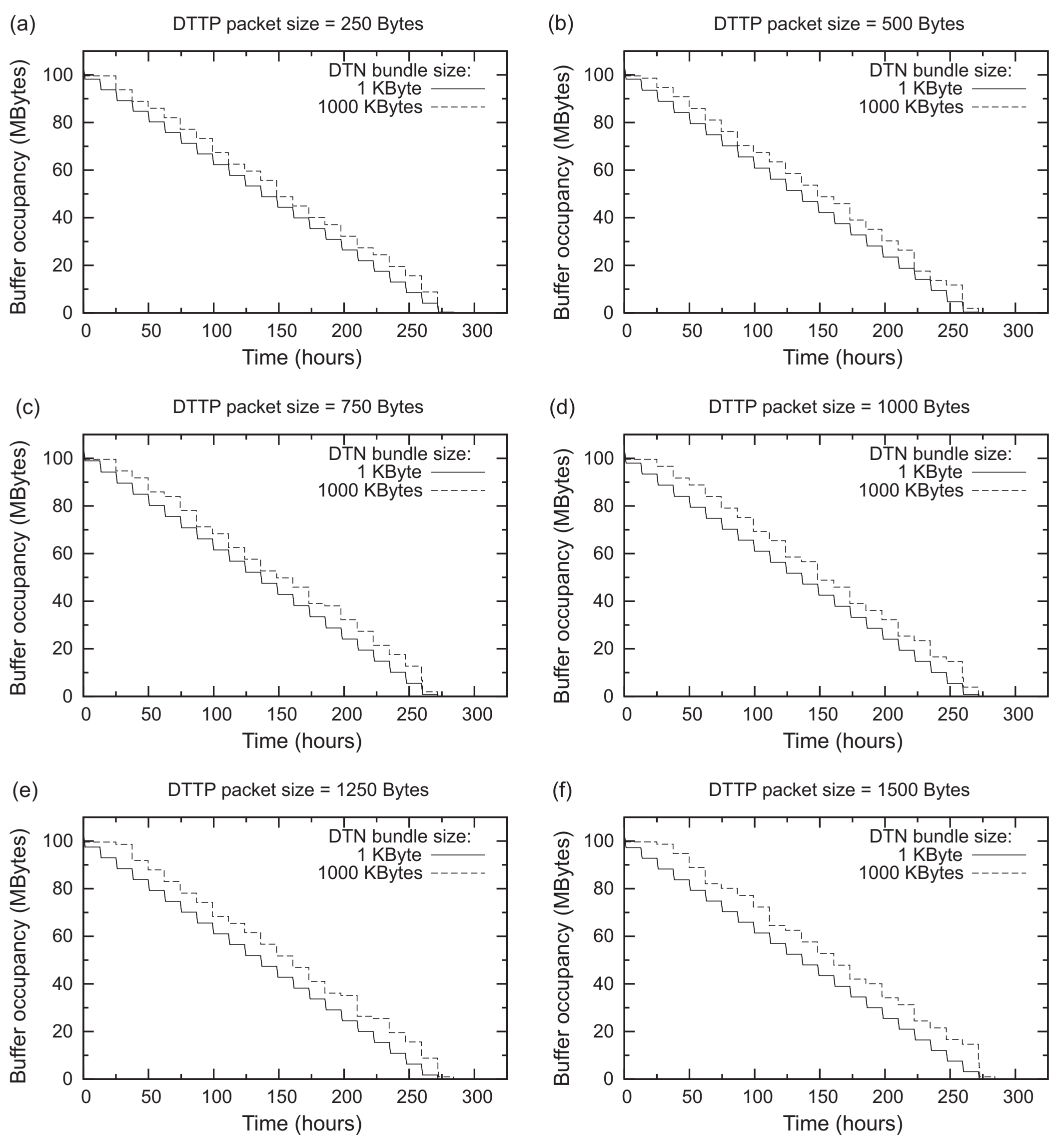

Fig. 10. Storage requirements at the sender.

the DTTP layer ultimately conditions sender's overall transmission effort.

5.3. Storage requirements and buffer space release rate

In Delay-Tolerant Networks, storage capacity on edge and intermediate nodes essentially forms a critical resource, that needs to be protected and carefully managed. DTN messages often need to wait on nodes for significant amount of time, until a new contact opportunity takes place. Faster release of occupied memory resources serves many roles: it can potentially support higher application data production rates on the node; it allows more efficient multiplexing of data flows through the node; and it improves energy efficiency, a desired property given the limited power availability aboard space probes. Sender buffer occupancy at the DTN bundle layer is presented in Fig. 10. The step-like graphs in the figure are justified, since buffer resources at the DTN sender get released according to bundle granularity. In other words, the entire DTN bundle is kept in memory, even if some parts of it have been successfully transmitted; the bundle is removed at once, after it has been transferred in its entirety and without errors to the next node. Additionally, the idiomorphic communication pattern in deep space (i.e., long propagation delays and even longer disconnection periods) can lead to successful transfer of multiple bundles, successively and during a relatively short period of time. Release of the associated memory brings quite steep steps in the relevant graphs.

Each subfigure in Fig. 10 concerns a specific DTTP packet size, whereas only the extreme values for DTN bundle size are examined each time, i.e., 1 kByte and 1000 kBytes. As expected, 1000-kByte bundles appear more memory-demanding than 1-kByte ones. Closer look at the graphs shows that the afore-mentioned difference in memory requirements further diverges with increasing DTTP packet size. However, Fig. 10 examines only a subset of the simulation set, and provides only visual representation on memory use. In order to minutely inspect how DTN and DTTP PDU sizes affect buffer occupancy, we utilize the Buffer Space Release Rate (BSR) index. This scalar metric essentially quantifies the rate by which memory resources are released. As evidently shown in Fig. 11, smaller DTN bundle sizes show clear advantage and release memory resources faster. Nevertheless, extremely small DTN bundles (e.g., see BSR graph for 1-kByte bundles) form an exception to the previous rule: for minimal bundle sizes, BSR rate is degraded as a direct consequence of considerable increase in DTN Protocol Overhead Ratio. In addition, the curved form of BSR lines reveals that intermediate values of DTTP packet size (ranging from about 500 to 1000 Bytes) perform better than smaller or larger DTTP packets, regardless of DTN bundle size. Memory-wise, best performing interaction among DTN and DTTP occurs for 5-kByte bundles and 750-Byte DTTP packets (BSR = 0.554), whereas most memory-demanding case is the combination of 1000-kByte bundles and 1500-Byte DTTP packets (BSR = 0.457).

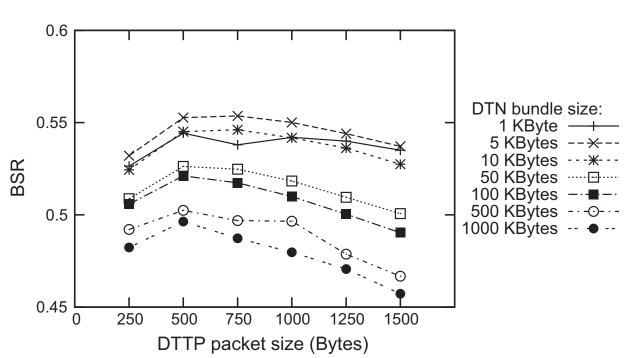

Fig. 11. Buffer Space Release Rate (BSR) at the sender.

Table 4
File delivery time minus file delivery time of best-performing case in the set (in minutes) for various bit error rates. DTN bundle size is 100 kBytes.

BER	DTTP packet size (Bytes)					
	250	500	750	1000	1250	1500
10^{-5}	1539.11	1479.98	1480.01	1500.15	1525.20	1538.68
10^{-6}	1480.02	59.53	38.23	28.71	23.85	1.63
10^{-7}	1479.96	55.12	31.39	0.03	0.03	0.00

Table 5
Sender's Extra Transmission Effort percent (XTE (%)) for various bit error rates. DTN bundle size is 100 kBytes.

BER	DTTP packet size (Bytes)					
	250	500	750	1000	1250	1500
10^{-5}	14.926	10.192	10.228	11.265	12.911	14.890
10^{-6}	12.854	6.387	4.558	3.741	3.325	3.141
10^{-7}	12.667	6.008	3.972	2.971	2.412	2.050

5.4. Impact of error rate

Last but not least, bit error rate plays a significant role in transport segmentation strategy. Due to space limitations, we present only a subset of relevant results, which involve DTN bundles of 100 kBytes. Table 4 lists file delivery time for various bit error rates and DTTP packet sizes. Apparently, higher error rates induce greater delays and possibly require extra transmission rounds. Sender's extra transmission effort (XTE) for various error rates and DTTP packet sizes is shown in Table 5. Interestingly, the observed pattern for bit errors at the order of 10^{-5}, where intermediate DTTP packet sizes (namely, 500–750 Bytes) perform better in terms of XTE (see Table 3), does not seem to hold for lower error rates (at least for the DTTP packet sizes under examination). Indeed, when BER is either 10^{-6} or 10^{-7}, sender's extra transmission effort tends to decrease as DTTP packet increases in size.

6. Requirements and design considerations of a transport adaptation scheme for packet size determination

A generic approach to transport segmentation policy in DTNs need not be protocol-specific. The main advantage of a protocol-agnostic solution is that it does not require additional header fields in protocols it operates upon. Next, we discuss a heuristic method for dynamic packet size determination based on network measurements.

In the previous section, we investigated DTN performance under the scope of synergistically setting PDU size at the DTN bundle and transport protocols. As seen in our findings, protocol overhead ratio at those layers and bit error rate combinatively affect various performance metrics, including but not limited to file delivery time, extra transmission effort at the sending side, and release rate of occupied buffer space. A *transport adaptation scheme* that dynamically adjusts PDU size requires:

- Selection of some *performance metric* that we wish to maximize. Different metrics may be selected by spacecraft to serve different needs, and a given spacecraft may switch to another metric, when different purposes need to be fulfilled. For example, a martian rover's default metric may target minimization of file delivery time to the next node, but when available energy resources become scarce, it might opt for reduction in wasted transmission effort (see discussion on XTE index in Section 4.2).
- Recording of *performance history* on a per-link basis. The goal is to record how the selected metric performs under various combinations of DTN bundle and transport packet sizes. To obtain

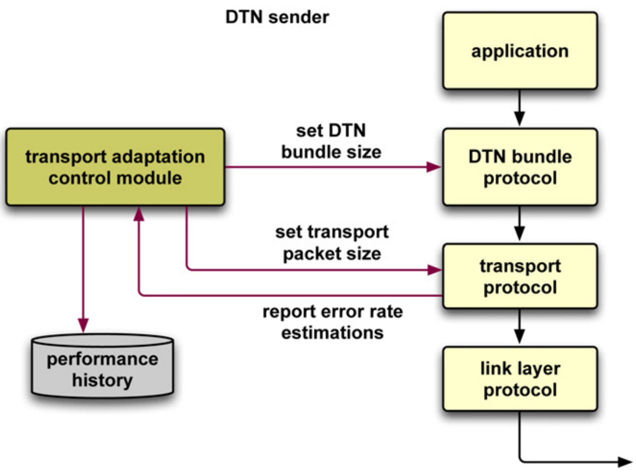

Fig. 12. Transport adaptation scheme.

the records, a respective algorithm is supplied with the DTN bundle size range [$bundle_{min}$, $bundle_{max}$], the transport packet size range [$packet_{min}$, $packet_{max}$], and relevant intermediate steps for those PDU ranges. Then, the algorithm iteratively picks all possible combinations for PDU sizes at the DTN and transport layers, and records performance across a predefined and carefully selected time period. Given a fixed or relatively stable error rate on the link, performance records of that form suffice. Otherwise, error rate should be considered as another variable in the set of performance records. As can be found in the literature, various methods/layers can serve to estimate error rate on the channel. We include an error rate estimator inside DTTP that relies on the acknowledgment history.

- A *control module* at the sender that interfaces with the DTN and transport layers. The module is responsible for executing the algorithm across all available space links originating from the sender. It stores locally and periodically updates the performance history, does not involve much computation, and dynamically adapts DTN and transport PDU sizes across each link.

The proposed transport adaptation scheme is shown in Fig. 12. As seen in the figure, the transport layer implements the error rate estimator, and provides error reporting services to the *transport adaptation control module*. The latter can adjust both the DTN and the transport PDU size for subsequent bundles and packets. The adaptive mechanism runs locally at the sender without need for network support.

Certain networking environments may favor success of the proposed transport segmentation technique while others not. Space networks in particular could incorporate the mechanism as they exhibit characteristics such as the following: number of space nodes (beyond Earth orbit) is limited and not going to grow substantially in the foreseeable future; by extension, available space links are limited too; network conditions per space link (such as propagation delay and error rate) remain stable or fluctuate between certain ranges; and connectivity schedule is well-known in advance which simplifies network performance monitoring.

7. Conclusions

Protocol designers typically work strictly within the scope of their functional goal. However, this scope alone cannot guarantee optimal communication systems: a system relies on interactions among layers and involves trade-offs and overlaps that may develop conflicting behaviors among system entities. Here, we investigated the impact of segmentation policy from a system perspective. We evaluated various policies, highlighted the cost of overhead due to small packets, and observed those cases that appear appropriate for a synergistic DTN, transport and link layer approach. We concluded that this synergy may reduce significantly memory expenditure and increase communication performance. Indeed, using our DTTP protocol and our implementation of DTN on the Network Simulator ns-2, we highlighted the potential of an adaptive segmentation approach that relies on the observation of the dynamics of system parameters.

References

Bisio, I., Marchese, M., de Cola, T., November 2008. Congestion aware routing strategies for DTN-based interplanetary networks. In: IEEE GLOBECOM, New Orleans, LA.
Burleigh, S.C., 2008. Dynamic routing for Delay-Tolerant Networking in space flight operations. In: SpaceOps 2008, Heidelberg, Germany, May.
Consultative Committee for Space Data Systems, September 2003. TM Space Data Link Protocol, Recommendation for Space Data System Standards, CCSDS 132.0-B-1, Blue Book, Issue 1. CCSDS, Washington, DC.
Consultative Committee for Space Data Systems, September 2003. TC Space Data Link Protocol, Recommendation for Space Data System Standards, CCSDS 232.0-B-1, Blue Book, Issue 1. CCSDS, Washington, DC.
Consultative Committee for Space Data Systems, July 2006. AOS Space Data Link Protocol, Recommendation for Space Data System Standards, CCSDS 732.0-B-2, Blue Book, Issue 2. CCSDS, Washington, DC.
Consultative Committee for Space Data Systems, July 2006. Proximity-1 Space Link Protocol—Data Link Layer, Recommendation for Space Data System Standards, CCSDS 211.0-B-4, Blue Book, Issue 4. CCSDS, Washington, DC.
Consultative Committee for Space Data Systems, January 2007. CCSDS File Delivery Protocol (CFDP), Recommendation for Space Data System Standards, CCSDS 727.0-B-4, Blue Book, Issue 4. CCSDS, Washington, DC.
The Consultative Committee for Space Data Systems (CCSDS), The Official Web Site: http://www.ccsds.org/.
Cerf, V., et al., April 2007. Delay-Tolerant Networking Architecture, RFC 4838, Internet Engineering, Task Force.
de Cola, T., Ernst, H., Marchese, M., 2007. Performance analysis of CCSDS File Delivery Protocol and erasure coding techniques in deep space environments. Computer Networks 51 (October (14)), 4032–4049.
Gao, J.L., SeGui, J.S., 2004. Performance evaluation of the CCSDS File Delivery Protocol latency and storage requirement. In: IEEE Aerospace Conference, Big Sky, MT, March.
Korhonen, J., Wang, Y., March 2005. Effect of packet size on loss rate and delay in wireless links. In: IEEE Wireless Communications and Networking Conference, vol. 3, New Orleans, LA, pp. 1608–1613.
Koutsogiannis, E., Papastergiou, G., Tsaoussidis, V., 2009. Evaluation of CCSDS File Delivery Protocol over Delay Tolerant Networks. In: 5th International Workshop on Satellite and Space Communications (IWSSC 2009), Sienna-Tuscany, Italy, September.
Lee, D.C., Baek, W., 2004. Expected file-delivery time of deferred NAK ARQ in CCSDS File-Delivery Protocol. IEEE Transactions on Communications 52 (August (8)).
Modiano, E., 1999. An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols. Wireless Networks, 279–286.
NASA's Mars Exploration Rover Mission, web site: http://marsrovers.jpl.nasa.gov/home/index.html.
The Network Simulator ns-2, web site: http://www.isi.edu/nsnam/ns/.
Papastergiou, G., Psaras, I., Tsaoussidis, V., 2009. Deep-Space Transport Protocol: a novel transport scheme for space DTNs. Special Issue of Computer Communications on Delay and Disruption Tolerant Networking 32 (October (16)), 1757–1767.
Ramadas, M., Burleigh, S., Farrell, S., September 2008. Licklider Transmission Protocol—Specification, RFC 5326, Internet Engineering Task Force.
Samaras, C.V., Tsaoussidis, V., 2010. Design of Delay-Tolerant Transport Protocol (DTTP) and its evaluation for Mars. Acta Astronautica 67 (October–November (7–8)), 863–880, doi:10.1016/j.actaastro.2010.05.025.
Scott, K., Burleigh, S., November 2007. Bundle Protocol Specification, RFC 5050, Internet Engineering Task Force.
Sung, I.U., Gao, J.L., June 2006. CFDP performance over weather-dependent Ka-band channel. In: 9th International Conference on Space Operations (SpaceOps), Rome, Italy.
Tarello, A., Gao, J.L., Modiano, E.H., March 2006. Energy efficient transmission scheduling over Mars proximity links. In: IEEE Aerospace Conference, Big Sky MT.
Wang, X., Yin, J., Agrawal, D.P., March 2005. Effects of contention window and packet size on the energy efficiency of wireless local area network. In: IEEE Wireless Communications and Networking Conference, New Orleans, LA.
Wang, R., Horan, S., Chandrasekaran, R., 2006. An experimental investigation of cross-layer optimal packet size in space Internet. International Journal of Satellite Communications and Networking 24 (November (6)), 561–577.
Wang, R., Wang, T., Wu, X., November 2009. Bundle Protocol (BP) over Licklider Transmission Protocol (LTP) for cislunar communications. In: IEEE GLOBECOM, Honolulu, Hawaii.

Wu, D., Ci, S., Sharif, H., Yang, Y., 2007. Packet size optimization for goodput enhancement of multi-rate wireless networks. In: IEEE Wireless Communications and Networking Conference, Hong Kong, March.
Yin, J., Wang, X., Agrawal, D.P., 2004. Optimal packet size in error-prone channel for IEEE 802. 11 distributed coordination function. In: IEEE Wireless Communications and Networking Conference, Atlanta, GA, March.

Christos V. Samaras received a B.Sc. in Geology from Aristotle University of Thessaloniki, Greece in 2000, and a M.Sc. in Information Systems from Northeastern University, USA in 2002. He has worked with the Cultural and Educational Technology Institute, Greece from 2004 to 2005. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece. His research interests lie in the area of transport protocols, deep-space communications, Interplanetary Networking, and Delay-Tolerant Networking.

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