Acta Astronautica 67 (2010) 863–880

Contents lists available at ScienceDirect

Acta Astronautica

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

Design of Delay-Tolerant Transport Protocol (DTTP) and its evaluation for Mars

Christos V. Samaras *,1 , Vassilis Tsaoussidis

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

ARTICLE INFO

Article history:
Received 20 November 2009
Received in revised form
5 March 2010
Accepted 30 May 2010
Available online 20 June 2010

Keywords:
Delay-Tolerant Network architecture
Interplanetary networking
Space Internetworking

ABSTRACT

We present Delay-Tolerant Transport Protocol (DTTP) which enables fast and reliable communications in deep space. The proposed protocol regularly measures its performance through available paths towards destination, and develops its routing strategy accordingly. We evaluate DTTP performance across a wide range of network scenarios that pertain to Mars-to-Earth distances. Based on simulation results, we show that: (i) Delay-Tolerant Networking does not necessarily imply belated communications; (ii) DTTP can efficiently support reliable data transfers in challenged networks; and (iii) dynamic routing is feasible in space.

1. Introduction

Space communications enter a new era, where space nodes (such as spacecraft, rovers, robots, devices, etc.) and ground stations on Earth will get interconnected and potentially attached to the terrestrial Internet. Applicable architectures for this new space infrastructure should be able to cope with extreme network characteristics. Excessive delays and network disconnections, in particular, call for new approaches that rely on asynchronous interactions coupled with network data storage for lengthy periods of time.

Towards the interplanetary Internet, the so-called Delay-Tolerant Network (DTN) architecture has gained broad acceptance among research community and space agencies, worldwide. Growing interest for DTN in space is underlined by ongoing standardization efforts under the auspices of the Consultative Committee for Space Data Systems (CCSDS) [1] and development of experimentation platforms for testing this network system, such as: NASA JPL's Deep Impact Networking Experiment on board the EPOXI spacecraft [2] (which demonstrated DTN operation in space), and ESA's DTN testbed deployment project [3,4].

Current schemes for space communications are insufficient, since they involve paths that are manually selected, require constant human intervention, and prohibit automation and sharing of space network resources. Moreover, delay-tolerant platforms (such as DTN) still lack properties or complementary protocols to enable efficient data transfers and accommodate various space application needs. As a consequence, a common misunderstanding about DTN services is that delay tolerance gets often misinterpreted as belated data transports. We argue that Delay-Tolerant Network systems can be designed with fast and efficient data communications in mind, given of course the limitations imposed by contact opportunities and long propagation delays. Indeed, delay-tolerance combats network interruptions, and gradually and steadily forwards data closer to destination (in terms of distance, delivery time as affected by connectivity schedule, or any other aspect). Even if no end-to-end path exists at any specific moment, store-and-forward procedures (especially when relied upon non-volatile memory) favor Delay-Tolerant Network performance. Also, we consider that DTN can potentially translate into more connectivity time: prompt

* Corresponding author. Tel.: +30 2541079553; fax: +30 2541079576.
E-mail addresses: csamaras@ee.duth.gr, christos.samaras@gmail.com (C.V. Samaras), vtsaousi@ee.duth.gr (V. Tsaoussidis).
[1] Current address: Smyrnis 5, 67100 Xanthi, Greece.

0094-5765/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.actaastro.2010.05.025

retransmission of lost data; exploitation of alternative forwarding paths in parallel; and adaptive routing based on connectivity maps and network measurements, are only some of the directions to boost performance of delay-tolerant applications and transform DTNs into fast and responsive networks.

We propose Delay-Tolerant Transport Protocol (DTTP) to serve efficient data transports in challenged space networks. DTTP complements delay-tolerant architectures by inspecting network performance and providing reliable and relatively fast transports [5]. We evaluate DTTP over a spectrum of typical-to-extreme network conditions. Our evaluation plan concentrates on network properties and performance in deep space scenarios, and seeks answers to open questions about challenged networks. What performance trade-offs may be exploited in such environments? How can data delivery time be reduced? Can network performance in Delay-Tolerant Networks be dynamically enhanced?

The remainder of the paper is organized as in the following. In Section 2, we provide an overview of Mars network as a synthesis of current and future Martian missions, and lay strategic technology directions and networking practices that pertain to deep space communications. Next, in Section 3, we present currently deployed or proposed protocols and approaches for space communications. In the following Section 4, we discuss DTTP properties, and elaborate on DTTP implementation details. We describe our network scenarios and relevant simulation results in Section 5. Finally, in Section 6, we conclude the paper and refer to future work.

2. What is changing in space networking?

Mars exploration holds a significant share of space exploration. Life presence, surface environment conditions, subsurface ice and water, geological composition, and exobiology, all comprise fundamental objectives that will yield a better understanding of the evolution and habitability of the red planet and the Earth, by extension. Currently operational missions to Mars include the orbiters: Mars Odyssey, Mars Express, and Mars Reconnaissance Orbiter; and the twin Mars Exploration Rovers (MERs): Spirit and Opportunity. High radiation levels and large temperature changes of space, impose computation constraints to planetary vehicles. The Mars Exploration Rovers have a 20 MHz RAD6000 CPU with 128 Mbytes of DRAM and 256 Mbytes of flash solid state storage. Rovers exhibit also limited communication capabilities for direct-to-Earth connections. MERs can return data direct to Earth using their X-band transmitters at a maximum rate of 28.4 kbits/s, which is an unachievable rate most of the time. However, relay operations via Odyssey orbiter have proved very successful: MERs can send data to Odyssey at either 128 or 256 kbits/s via UHF links, and that has become the preferred data forwarding path.

Among future space missions to Mars, we note Mars Science Laboratory, ExoMars, and Mars Sample Return mission. Given the desire to lessen space mission costs, space agencies engage in joint missions (e.g., the Mars Sample Return project), and adopt practices for resource sharing in space.[1] At the same time, Mars orbiters belonging to NASA or ESA form the next-hop network nodes for current (and future) missions on Mars surface, including landers, rovers, etc. Each orbiter can serve multiple missions belonging to the same or some different space agency.

Mars Science Laboratory is a NASA rover to land on Mars, that is scheduled to be launched at the end of 2011. The rover's autonomous capabilities are similar to those of MER vehicles [6], and its computer uses a 200 MHz RAD750 CPU with 256 Mbytes of DRAM and 2 Gbytes of flash memory. ExoMars mission is under development by European Space Agency (ESA) in possible collaboration with NASA, and Mars Sample Return is a joint project of NASA and ESA. Each of those two missions will deploy an orbiter, a lander, and a rover. The descent module of Mars Sample Return will also encompass an ascent vehicle that will collect martian soil and, for the first time, return it back to Earth for detailed analysis [7]. The orbiters will be able to operate as data relay satellites and might serve other missions as well.

ExoMars' scientific data will be transmitted back to Earth via Mars orbiter relay satellites. Transmission between the rover and the relay satellite will be based on UHF links, while data transmission between the relay satellite and ground stations on Earth will use X-band. Total data volume transmitted per day is expected to be approximately 120 Mbits. The ExoMars rover will also have a direct to and from Earth communication capability in X-band at the order of hundred bits per second, which will be possibly exploited to update the rover's daily plan of tasks on Mars each local morning [8].

Communications are of critical importance to space exploration. Doubts are raised whether current radio-frequency (RF) communications could accommodate complex missions with high data volumes, such as the Mars Sample Return mission. Studies are under way to consider the transition from RF to optical communication-systems, or the coexistence of those to support deep space missions. Indeed, optical communications provide higher data rates, while consuming less energy and adding less mass and volume to the spacecraft [9]. While spacecraft optical communications have been demonstrated in Earth orbit [10], they have not been deployed yet at interplanetary distances. As an alternative to optical systems, new high-rate communication techniques are investigated. These methods include: very low complexity error correction coding to improve Ka-band link availability for gigabit per second (Gbps) links; software configurable radios that adaptively mitigate amplifier distortions throughout the life of long-duration missions; and

[1] For instance, ESA and its Russian counterpart, Roscosmos, signed an agreement in August 2009 to cooperate on two Mars exploration projects: ESA's ExoMars will use Russia's Proton rocket and buy Russian parts for the ExoMars rover power supply system, whereas Roscosmos' Phobos–Grunt mission (which is a Russian sample return mission to Phobos, one of the moons of Mars) will use ESA's ground communication facilities for data communication to and from Earth.

integrated wideband array combiner and telemetry receivers for bandwidth-efficient signals [11].

Current and future Mars and deep space missions, in general, call for increased storage and communication resources, enhanced connectivity, and multi-path options for reliable data transfers. Future science data requirements dictate high communication rates, as shown in Fig. 1 taken from NASA JPL Strategic Directions 2009 report [11]. Prior to deploying manned missions to Mars, a network infrastructure needs to be constructed and established atop standard communication protocols and interfaces. Indeed, resource sharing and cross-support interoperability among different space missions and across space agencies can increase data return rates, reduce mission operating costs, drive complex space missions, and supply the indispensable communication architecture even for human visits to Mars. The future interoperable network architecture should also be able to accommodate each space agency's policies. For instance, a space mission might prefer to defer data transmission until contact links via its own assets become available, even if there exist a shorter path using some other agency's resources. Or an agency might decide that routing its high-priority data via its space nodes will delay too long, and compromise with another agency's satellites or ground stations on Earth. Inter-agency service level agreements, flexibility in selecting data paths, and policy-based routing could serve such cases and preserve individual policies of space missions. To summarize, desired properties for deep space communications include, but are not limited to, the following: (i) automatic store-and-forward data forwarding; (ii) operation through multiple network hops; (iii) transfers that can span multiple ground stations on Earth; (iv) autonomous networking potentially supported by dynamic routing procedures; (v) high data rate transfers that effectively utilize contact opportunities and minimize data delivery time; (vi) secure transmissions; and (vii) multiple quality of service levels.

3. Related work

Reliable and efficient communications in space largely count on manually scheduled, one-hop connections, which require costly human operations and prevent resource sharing and networking in space. Research community is actively investigating architectures and protocols that will transform space assets into an automated network infrastructure. Indeed, recent practice of communicating data from NASA's rovers on Mars (Spirit and Opportunity) through relay-satellites orbiting the planet [12], has proved advantageous over direct-to-Earth communications: energy resource of rovers is better preserved and data return rates increase. To this end, various approaches have been proposed and analyzed for data exchange in space. In the following, we refer to candidate technologies for interoperability in space (namely, Space Packets, IP in space, and Delay-Tolerant Networking); protocols that provide reliable data transfer services (either across single-hop connections or end-to-end solutions); and adaptive or dynamic routing schemes for Delay-Tolerant Networks.

Within the Open System Interconnection Reference Model (OSI Seven Layer Model), Space Internetworking requirements imply that routing across different space agencies and among ground segment, space nodes, and planetary networks cannot be offered at link layer (as is currently the case), but should extend to network layer or above. CCSDS Space Packet Protocol [13] has been proposed as the common communication layer for inter-agency data exchange, and has the advantage that the supporting ground and space backbone infrastructure is operational today. The downside is that CCSDS Space Packets have been designed for transfers across point-to-point connections or simple network topologies, and thus cannot meet the requirements of novel space network architectures.

IP in space has been proposed and extensively studied. Since IP comprises a mature technology in terrestrial

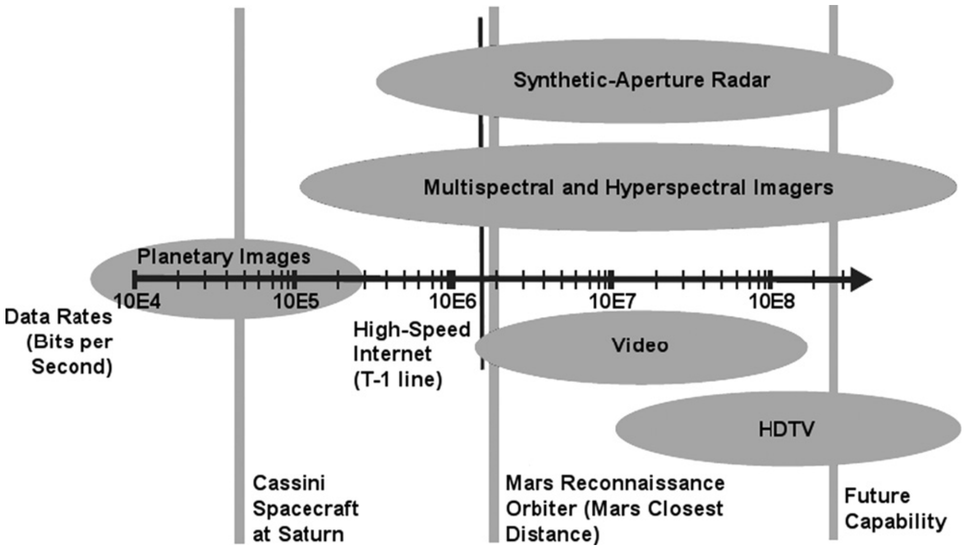

Fig. 1. High bandwidth communications required by future space applications.

networks and basically in the Internet, its adoption in Space was initially expected a natural choice for space data systems interconnection. Nevertheless, for its proper and efficient function, IP assumes that certain conditions are met, such as: well-connected links; permanent connectivity; and short round-trip delays (at the orders of seconds). Thus, IP could function effectively for missions up to Lunar distances only, or form IP-enabled island-like networks. Substantial modifications in current space infrastructure are needed in order to launch IP in space, and several aspects should be resolved (for example, routing tables construction, accountability, security) before its adoption in space missions.

Delay-Tolerant Networking has been conceived to support challenged networks: it addresses the technical issues of heterogeneous networks that may lack continuous network connectivity and/or suffer from long delays and high bit error rates [14,15]. DTN essentially employs store-and-forward message routing to overcome communication disruptions, and supports custody transfer of messages (also known as *bundles* in DTN terminology). In essence, custody transfer mechanism allows a source to delegate retransmission responsibility of bundles to next node on the path to final destination, and to resume the associated resources more quickly. Candidate protocol stacks for DTN in space can be seen in [16]. However, there still exist open issues for the deployment of DTN in space, such as: transport services that aim fast data transfer; relevant routing procedures that enhance network performance and minimize data delivery in space; bridging between lower layers protocols at waypoints, etc. Open research areas and problems with DTN Bundle Protocol are discussed in [17,18]. In the following, we present currently operational or proposed protocols and approaches that relate to Delay-Tolerant Networks or space networks in particular.

Deep-Space Transport Protocol (DS-TP) targets efficient and reliable communications in deep space. Among its features, we note the mechanism for redundant packets transmission (*Double Automatic Retransmission*), that serves to proactively protect against link errors. We have evaluated DS-TP in [19] and shown relevant performance gains.

CCSDS File Delivery Protocol (CFDP) [20] is capable of transferring files to and from spacecraft mass memory. In its simplest form, CFDP *Core Procedures* provide file copy services over a single link. When source and destination nodes do not connect directly, the protocol offers *Extended procedures* that perform multiple file copy operations across each link of the path to final destination. Route adaptation is not inherently supported by the protocol, and fully autonomous handover operations (e.g., ground station handover) should be established by system functions external to CFDP. Currently, CFDP store-and-forward procedures require all parts of a file to follow the same path to destination.

Licklider Transmission Protocol (LTP) [21] provides retransmission-based reliable transfers over single-hop connections. LTP is designed to serve as a DTN convergence layer protocol over long and/or often disconnected links, such as those encountered in the interplanetary network setting. LTP supports both reliable and unreliable data transmission.

Contact Graph Routing [22] is a space-oriented routing system that inserts dynamic path selections. It is based on DTN, assumes complete knowledge of scheduled communication contacts, and computes routes on the principle of bypassing oversubscribed nodes. Contact Graph Routing is appropriate for networks where changes in connectivity are planned and scheduled, rather than predicted or discovered.

Our proposal is the Delay-Tolerant Transport Protocol (DTTP). Our contribution is mainly to introduce dynamic characteristics into space data transports to enhance efficiency. DTTP differs from the aforementioned protocols primarily in its reliable end-to-end transfer services, with innate support for route adaptation. Also, DTTP offers multi-hop data transfer services through potentially multiple paths, even for the same DTTP session. In contrast to DTN Bundle Protocol's data unit (i.e., the *bundle*) that may occasionally increase considerably in size,[2] DTTP's protocol data unit (i.e., the *packet*) has smaller size and results in more efficient in-network storage utilization, especially when link errors are present. Within the evolving space network (corollary of international collaboration among space agencies), DTTP can offer flexibility in data routing, distributed in-network data storage, and effective transmissions to support growing needs of space applications. We describe DTTP in detail in the following chapter.

Recent advances in communication protocols/architectures and Space Internetworking services can be found in the following references. The Space Internetworking Strategy Group (SISG), chartered by the Inter-agency Operations Advisory Group (IOAG) [24], has reached some conclusions on current state and the future of Space Internetworking (see [25]). The Delay Tolerant Networking Working Group (SIS-DTN), which falls under the Space Internetworking Services (SIS) Areas of the CCSDS, describes in [16] the rationale, scenarios, use cases, and requirements for a proposed Delay-Tolerant Networking (DTN) service targeted at the Space Internetworking environment. Also, an operations overview for Mars mission interoperability and relevant communication protocols to be used in Mars end-to-end operations are provided in [26].

4. DTTP protocol: features and implementation

4.1. DTTP features

Delay-Tolerant Transport Protocol (DTTP) supports reliable communication in challenged environments. Focus is placed on space networks, though it may well apply to other network settings too. DTTP is conversational in nature as discussed in the following. However,

[2] DTN bundles of 150 Mbytes have been tested and sent by Low Earth Orbit UK-DMC imaging satellite (see [23]), while DTN bundle size for deep space communications is likely to take values in the order of 10s or 100s of kbytes.

interactions between sending and receiving nodes (may) take place asynchronously in a delay-/disruption-tolerant way, depending on contact opportunities between nodes. Practically, DTTP relies on an open-loop model for transmission scheduling coupled with a closed-loop system for administration.

Via management procedures, each node gets assigned a single *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. Routing procedures, that determine forwarding paths, are discussed later in this section. Data traffic flows in one direction only, from source to destination node, while data-acknowledgment traffic follows the opposite direction. The structure of DTTP protocol data unit (simply called *packet* from now on) is shown in Fig. 2. It consists the *Header* and *Data* segments, which are 28 bytes and variable length, respectively.

The fields of DTTP packet header are explained in the following:

- *Source Node ID* identifies DTTP source entity and, similarly, *Destination Node* identifies (final) DTTP destination entity. *Current Custodian Node ID* points to the node currently responsible for reliable delivery of the packet in question. Initially, the source node serves as the current custodian, but as DTTP packets move inside the network, other nodes may acquire custody of the packets. The notion of custody transfer is explained in more detail later in this section. Node ID takes up 32 bits.
- *Sequence Number* holds the sequence number of the first byte in the packet's data payload. Numbering starts at 0, and denotes the first byte of *total application data*. ("Total application data" refers to overall data to be transferred within a specific DTTP session.)
- 4-byte *Timestamp* value is set to DTTP packet creation time at Source Node ID and remains constant. Timestamp together with Sequence number serve to uniquely identify a DTTP packet belonging to a certain end-to-end DTTP session. Timestamp also protects against wrapped sequence numbers in cases where application data exceeds sequence number capacity (namely, for files or data streams over 4 Gbytes, which is the upper limit of 32-bit sequence number space).
- *Length* marks the size of data payload (in bytes) that the DTTP packet carries. Length value coupled with Sequence Number determine which portion of total application data is contained in the DTTP packet.
- 16-bit *Checksum* is used for error-checking of DTTP header and data, as errors might be introduced during storage or transmission. Every time a header field is updated (e.g., Current Custodian Node changes or DTTP packet gets fragmented), the Checksum is recomputed.
- *Session ID* gets initialized at the beginning of a DTTP session by the Source Node, remains constant for the duration of the session, and may be used to multiplex/demultiplex different DTTP sessions between the same pair of source-destination nodes. Session IDs are locally assigned and administered. The 3-tuple {Source Node ID, Destination Node ID, Session ID} uniquely identifies a DTTP session. This 3-tuple forms the demultiplexing key for the DTTP protocol. Session ID is 10 bits in length, which allows for 1024 distinct and concurrent, end-to-end DTTP sessions from Source Node to Destination Node (and another 1024 sessions between the same nodes for data flowing the reverse direction).
- In case that a DTTP packet gets fragmented, *Fragment Offset* refers to 8-byte blocks in the initial (that is, *before* fragmentation) DTTP data segment, and specifies the offset of the fragment relative to the unfragmented DTTP packet data.
- The *FR* flag bit signifies whether the DTTP packet is fragmented or not. When fragmentation occurs, the *FR* boolean is set, and fields *Length* and *Fragment Offset* get their values updated. Also, the last-fragment flag (*LF*) is either set to 1 for the last fragment or to 0 for all previous fragments.

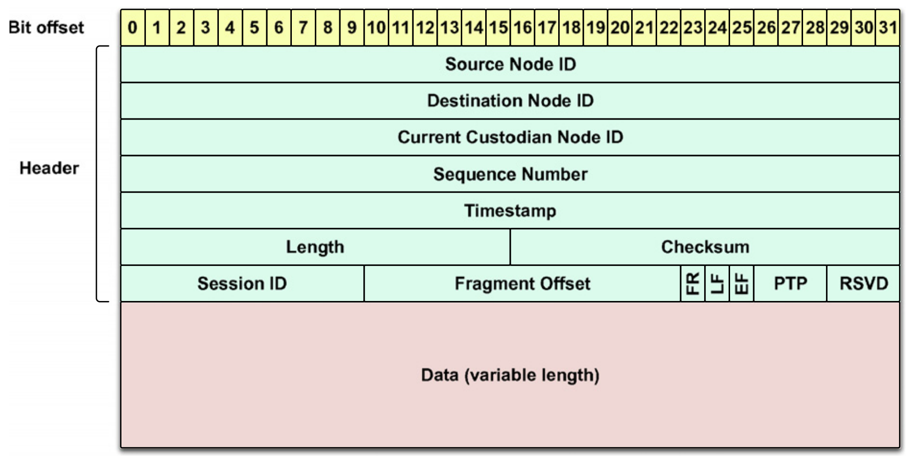

Fig. 2. DTTP packet format.

- The last DTTP packet in a DTTP session must set the *EF* (end-of-file) flag to '1'; for all other DTTP packets in the session, this flag must be set to '0'.
- The 3-bit *Packet Type* field (PTP), as the name implies, defines DTTP packet type. It is '000' for data packets and '001' for acknowledgment (ACK) packets. Remaining values are reserved for other packet types, that can provide special services (e.g., data encryption), control DTTP session (e.g., packet to request "session termination"), or packets that include extension blocks for specialized functionality.
- The remaining 3 bits in the DTTP packet header (*RSVD*) are reserved for future use.
- The *Data* segment holds pure application data in case of data packets (packet type '000'), or acknowledgment information in case of ACK packets (packet type '001'). For other packet types, data segment semantics are to be defined accordingly.

DTTP operation is depicted in Fig. 3. The basic features of DTTP protocol include:

(i) *Reliability*: DTTP provides reliable transport services. DTTP agents run on every node across multi-hop paths, and reliability is provided distinctly on each link of the path. DTTP supports cumulative and selective acknowledgments to ensure complete transfer of data. These acknowledgment methods are used collaboratively in DTTP ACK packets. They rely on the fact that the original byte sequence of total application data, is numbered and preserved in transmitted DTTP packets throughout the DTTP session (see *Sequence number* and *length* fields of DTTP packet format). Cumulative acknowledgments acknowledge data application bytes up to the reported sequence number, whereas selective acknowledgments acknowledge blocks of contiguous application data bytes. Retransmission timer, in particular, is set to $2D_{prop}+D_{admin}$, where D_{prop} is the one-way propagation delay of the respective link and D_{admin} is the administrative delay that reflects the queuing delay on-board spacecraft plus the transmission delay of DTTP data. After a retransmission timer expires, DTTP gives precedence to pending retransmissions, and afterwards resumes transmission of unsent data. DTTP suspends retransmission timer for packets sent during the last $2D_{prop}$ seconds of a communication contact. Retransmission timer is resumed when a new communication contact begins. The reason is that, even if those last packets arrive successfully at the receive node, their acknowledgment packets cannot arrive at the send node at all, since the communication contact would have ended by that time. Consequently, the respective acknowledgments also await transmission at the receiver until a new communication contact takes place. This algorithm is based on the assumption that the communication schedule is known to the DTTP nodes in advance.

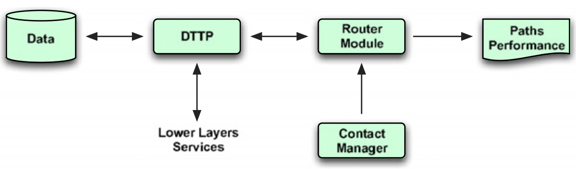

Fig. 3. DTTP operation.

(ii) *Custody transfer*: The Protocol Data Unit (PDU) of DTTP is a packet. Relying upon store-and-forward procedures, DTTP employs persistent storage of its packets inside nodes they traverse. Persistent store helps combat network interruptions and data can survive even after a spacecraft system restart in critical situations. Custody transfer process delegates reliable transfer responsibility from one node to the next on the path to final destination. Once a packet is received at an intermediate DTTP receiver and its custody is accepted, the sender can delete it from its buffers upon receipt of the relevant acknowledgment. This feature serves mainly a two-fold role. First, lost/corrupted packets get detected and retransmitted much faster on the basis of each link rather than end-to-end, especially when disconnection periods and delays increase in value. Second, smoother storage occupancy is usually achieved as packet load gets balanced and stored in participating DTTP nodes (including intermediate ones) on the path towards the ultimate DTTP receiver. A DTTP packet's current custodian is indicated in the relevant field of the packet (*Current Custodian Node ID*). DTTP employs custody transfer for all DTTP packets. Therefore, DTTP acknowledgment packets essentially provide two functions at the same time: inform about successful receipt of data, and perform custody acceptance of the relevant DTTP packets.

(iii) *Routing path adaptation*: DTTP design allows for manual or dynamic routing path selection. The former refers to how routing is typically performed in space communications nowadays; the latter is a future target for space agencies worldwide. In order to make efficient route selections, space nodes maintain state about upcoming communication contacts (start time, duration and transmission characteristics of contact, etc.). Depending on space missions priorities, several metrics could have been selected as a decisive factor for route adaptation. We have selected *data goodput*, which is defined as: size of *application data* successfully transmitted (and received by next node) during a communication contact divided by the transmission time (in bytes/s). *Application data* refers to data carried inside the data segment of DTTP packets. Successful transmission of DTTP packets is inferred upon receipt of the respective acknowledgments. Hence, goodput is computed locally at the DTTP sender, and no other subsequent node is required to report any special information to the sender. Selecting a path of higher goodput, enables a remote spacecraft to relatively fast offload its data. Thus, buffer space resources, which tend to be scarce and valuable in space, get freed up relatively quickly. DTTP acts as an autonomous agent that builds its knowledge-base on routing paths efficiency, goodput-wise. We have implemented the adaptive routing mechanism locally at the sender-side (Fig. 3). Since no sender-receiver interaction takes place, this mechanism does not reflect on any DTTP header field.

Given the problem of selecting one path out of two (or more), DTTP's general tactic breaks down to these tasks:

- forward data through all paths consecutively (*probing phase*);
- measure achieved goodput on the paths (construct *knowledge-base*);
- select path of maximum goodput performance;
- update performance of current path, periodically (e.g., once per contact);
- select next path in the descending-performance group of available paths, when performance of current path degrades; and
- (optionally) select available paths periodically and iteratively to update their performance (check for changes in paths' efficiency).

Records of the form {*path, performance*} are maintained at the router module of DTTP nodes, so that performance across available paths is recorded and regularly updated. For our proposed routing policy, performance of a path refers to the size of data successfully injected via this path, is bound to some defined time period (transmission round), and in essence reflects data goodput. Our path selection algorithm biases towards paths whose maximum performance recently recorded is high. Hence, mild performance degradations are tolerated and high-performance paths are expected to some degree to respond equivalently well in the near term. More specifically, performance of a path (P), in terms of data successfully transmitted, is calculated by the weighted mean of current performance ($cur\,P$) and maximum performance recorded ($\max P$):

$$P = W_{max\,P} \cdot \max P + (1 - W_{max\,P}) \cdot curP$$

A convenient value for the weight of maximum performance ($W_{max\,P}$) that we have found practical in our simulations is 0.3, for which the above formula becomes:

$$P = 0.3 \cdot \max P + 0.7 \cdot cur\,P$$

However, we have only justified this selection experimentally and further analysis is needed to finalize weights.

Different needs of space missions or applications might prefer a different route selection algorithm to the one we have proposed. In fact, we argue that this should be a configurable parameter that enables a space node/application (or even space network administration at a global level) to satisfy their specific requirements. Since DTTP operation is independent of routing schemes, a number of diverse routing algorithms can be implemented and installed onboard spacecraft. Selecting the most appropriate routing scheme could then fulfill different goals of space missions/applications.

The state transition diagram of DTTP sender is depicted in Fig. 4. When a communication link becomes available or unavailable, DTTP enters transmission mode or idle mode, respectively. Well-connected networks (e.g., the Internet) and conversational protocols (such as TCP) assume fast end-to-end message exchanges for their proper operation, and indication of even a single missing packet can trigger retransmission and even reduce sending rate. On the contrary, DTTP retransmission timer is set collectively for a group of DTTP packets (*massive retransmission timer*), and its expiration takes into account total delay (propagation, transmission, queuing, processing) for DTTP packets, as well as delay imposed by link disconnection. Upon timer expiration, retransmission of missing packets take precedence over transmission of new DTTP packets that await in the sending queue. Depending on network scenario and connectivity schedule, retransmission can occur either at the beginning of a communication contact (in cases when timer expires while the communication link is down) or during that communication contact.

Delay-Tolerant Networking (DTN) Bundle Protocol [15] has been proposed to provide message exchange in highly stressed environments; it sits at the application layer of some number of constituent networks, and can cope well with large delays or network disconnections. DTTP deployment in absence or presence of DTN Bundle Protocol is shown in Figs. 5 and 6, respectively. In the former case, a suitable network protocol is needed to interconnect spacecraft among each other and connect those to ground stations on Earth: we use IP in the figure just for demonstration purposes. (Actually, we have used IP in our simulations, though any other addressing scheme can be exploited.) In the latter case, DTTP provides reliable transport services to DTN and acts as DTN's convergence layer: DTTP agents send and receive DTN messages, also known as DTN *bundles*, rather than interfacing directly with the application. In this network setting: (i) DTTP routing computations can be integrated into and provided by DTN system implementation, i.e. routing decisions could be made on a higher level in the protocol stack; and (ii) DTTP key role involves efficient reliable data transfers, as DTN Bundle Protocol lacks such functionality. See [27] for a relevant discussion.

4.2. DTTP implementation

We have implemented DTTP on the Network Simulator ns-2 [28]. A major shift from conventional ns-2 functionality is that our implementation circumvents typical queue functionality: buffers do not build up similarly to how Internet routers work; instead, non-volatile storage resources are simulated, and data is read from storage, transmitted, and released from storage after acknowledgment arrives. Buffer takes the form of persistent memory to combat network disruptions and, as an effect, network congestion amounts to storage congestion. We evaluate DTTP over IP, where IP is used solely for node addressing and DTTP packet forwarding. Routing procedures are provided by the relevant module of DTTP, which

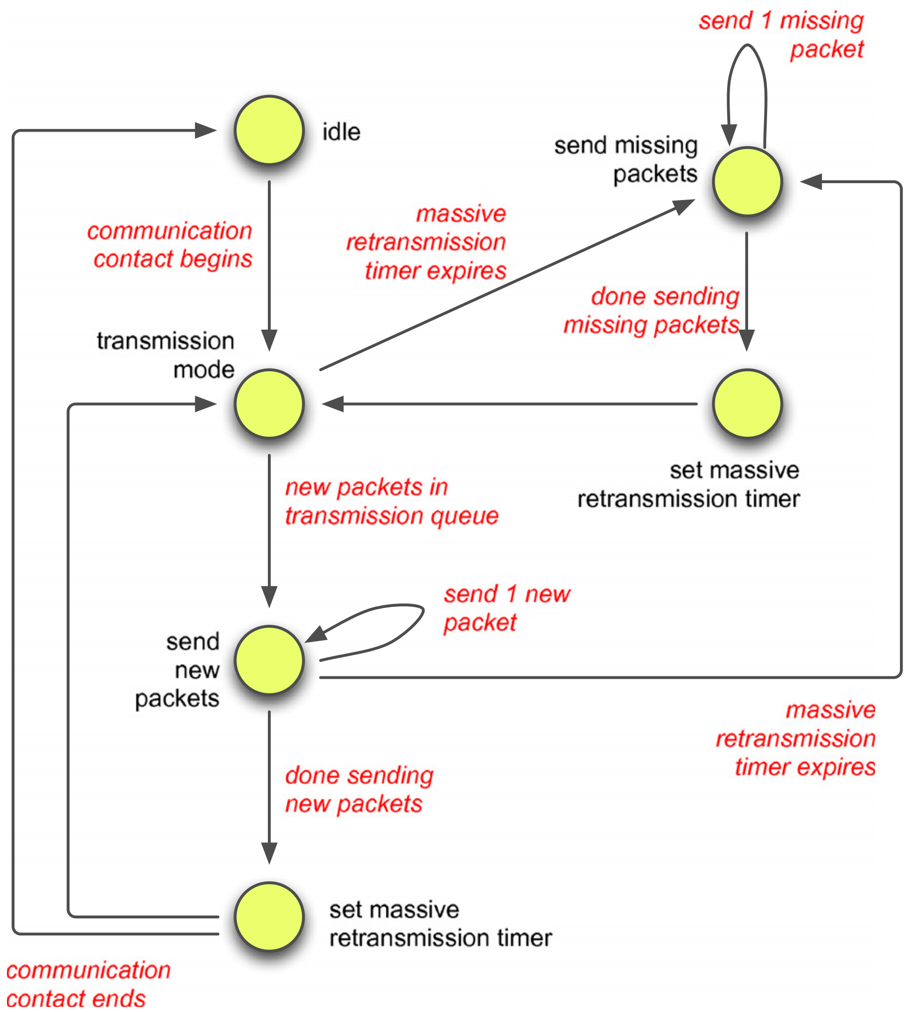

Fig. 4. DTTP state transition diagram.

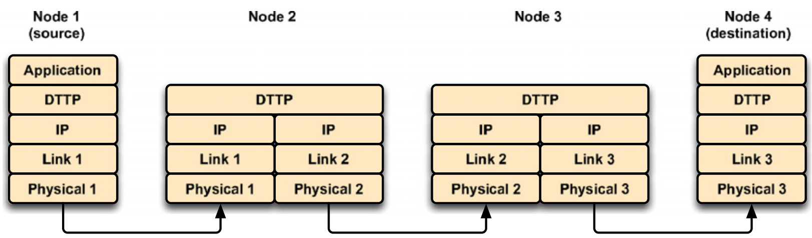

Fig. 5. DTTP deployed in absence of DTN Bundle Protocol.

measures paths performance (as described in Section 4.1) and dynamically selects a routing path. Whenever a DTTP node receives a packet, it checks for store availability. If storage space is available, custody of packet is accepted (i.e., the node becomes responsible for reliable transmission of packet to next node), and acknowledgment mechanisms report its receipt. Otherwise, the packet is deleted, and path-selection mechanism at the sender side is responsible for either retrying transmitting through the same path or selecting another route.

We have implemented in Network Simulator a reporting system that monitors DTTP operation at every node. Detailed records are kept during simulations. More specifically, this reporting module records: changes in buffer occupancy at each network node; network route selections; errors interrupting normal operation of DTTP; dated events of DTTP packets transmission, retransmission, receipt, and custody acceptance (deep inspection of network data); metrics to evaluate DTTP/network performance (e.g., total number of transmitted

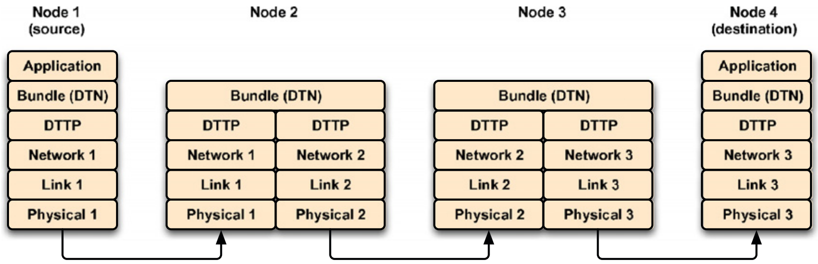

Fig. 6. DTTP deployed in DTN-based network.

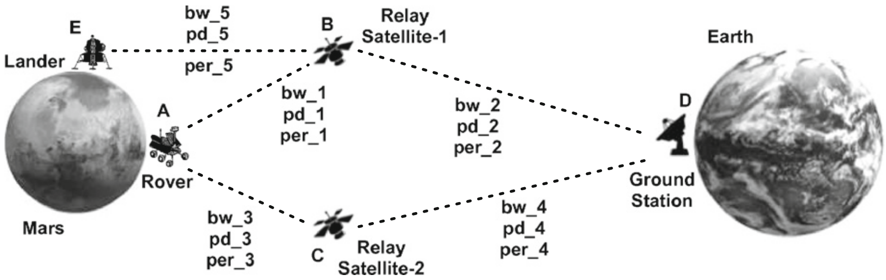

Fig. 7. Deep space network topology (bw=bandwidth, pd=propagation delay, and per=packet error rate.)

and retransmitted packets at DTTP senders, number of new packets and retransmitted packets received at DTTP receivers, number of packets dropped at intermediate nodes due to exhausted memory, etc.).

5. Simulation results and discussion

Under the aforementioned Mars-network framework, we have constructed our simulation scenarios. We consider a deep-space scenario, where a rover on Mars transfers data to a ground station on Earth via one or both of the relay satellites orbiting Mars (Fig. 7). (For simplicity, data to be transferred is referred collectively as one file.) Each of the 2 relay satellites needs 2 h to complete a total orbit around Mars. During each 2-h period (i.e., one *transmission round*), we assume that the rover's energy resources (on the average) allow it to pick one of the two satellites to forward data to, and not both satellites. So, the problem for the rover is simplified as to which relay satellite to select every 2 h, with the aim of fast data delivery to ultimate destination. However, depending on the simulation scenario, a lander on Mars may occasionally inject data into the network, thus affecting network dynamics.

DTTP is deployed at the space segment of the end-to-end connection, that also includes ground stations on Earth. In our simulations, we terminate the network topology at ground stations on Earth and omit the ground link between ground stations and Mission Operation Centers (or even subsequent recipient of space data, such as research institutes, universities, etc.). We assume, as is normally the case, that the delay burden lies in the space portion of the network, where extraordinary propagation delays and network disconnections are present, while the well-connected terrestrial network poses insignificant delays.

Table 1
Configuration of simulation parameters.

	Scenario I	Scenario II	Scenario III
bw_1 (kbits/s)	20	30	40
bw_2 (kbits/s)	50	50	50
bw_3 (kbits/s)	40	40	30
bw_4 (kbits/s)	50	50	50
bw_5 (kbits/s)	N/A	N/A	20
pd_1 (s)	0.016	0.016	0.016
pd_2 (min)	5, 10, or 20	5, 10, or 20	20
pd_3 (s)	0.016	0.016	0.016
pd_4 (min)	5, 10, or 20	5, 10, or 20	20
pd_5 (s)	N/A	N/A	0.016
per_1	0.1%	0.1%	0.1%
per_2	1% or 5%	1% or 5%	1%
per_3	0.1%	0.1%	0.1%
per_4	1% or 5%	1% or 5%	1%
per_5	N/A	N/A	0.1%
file size (Mbytes)	10, 20, or 50	10, 20, or 50	10, 20, or 50

Fig. 7 presents the deep space network topology used in our simulations. It comprises a rover (node A) and a lander (node E) on Mars, two relay satellites orbiting Mars (nodes B and C), and a ground station on Earth (node D). Also, administrative delay (D_{admin}) is set to 5 s without inducing any redundant retransmissions (see Section 4.1, DTTP Reliability). DTTP packet size is 1 kbyte. Propagation delay for the long-haul links between Mars and Earth (i.e., links BD and CD in Fig. 7) varies from 5 to 20 min, since the distance between the two planets varies as they move in their orbits. Configuration of simulations is shown in Table 1: network parameters include link bandwidth (*bw_*), propagation delay (*pd_*), packet error rate (*per_*), and size of data to be transmitted (*file size*). Space nodes' connectivity schedule is depicted in Fig. 8: an active link connection is denoted by a gray bar; otherwise, the link is disconnected.

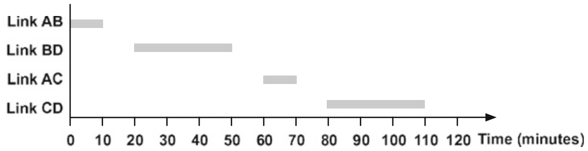

Fig. 8. Connectivity schedule.

This connectivity pattern repeats itself every 2 h, which equals satellites' orbiting period around Mars. We note that in order to construct a more realistic space network connectivity schedule, various other factors should be quantified as well, such as rotation of Mars and Earth, relay spacecraft orbit planes, latitude of Mars rovers, etc. However, despite the simplified network connectivity schedule we deploy, performance of protocols (examined in our simulations) is affected in a similar way, and thus valid results and relevant conclusions can be drawn.

Along with DTTP, we implemented a similar protocol, the "Static Protocol" (*SProt*), to serve as a reference protocol. SProt shares the same features with DTTP, with the only difference that SProt lacks dynamic routing functionality. Instead, SProt routes data through a predefined and fixed path, which is essentially the current approach in space communications. In regard to our space network topology (Fig. 7), *SProt*-1 manually selects path A–B–D as its routing path, runs on the rover and on each subsequent node towards the final destination (namely, nodes A, B, and D), and retains its static path selection until all data gets successfully received at the ground station on Earth (node D). Similarly, *SProt*-2 routes its data across the path A–C–D, runs on the respective nodes (nodes A, C, and D), and transmits its packets via the same network path until all its data gets successfully delivered. On the contrary, one DTTP session can exploit both paths (A–B–D and A–C–D), runs on all nodes (A, B, C, and D), and dynamically adjusts its path selection based on its route-selection strategy. However, as we have noted at the beginning of this section, we assume that energy resources of the rover (node A) do not permit it to exploit both relay satellites during one *transmission round*. In other words, during each transmission round (i.e., every 2 h), DTTP (running on the rover) should select to forward its packets to either the first relay satellite (node B) or the second relay satellite (node C).

Our evaluation plan was implemented on the Network Simulator ns-2 [28]. We assess the efficiency of our proposed protocol, DTTP, from the perspective of data delivery time, size of retransmissions, memory buffer occupancy, and responsiveness to changing network conditions. We evaluate the impact of: (i) unequal contact capacities across available paths; (ii) storage congestion at intermediate network nodes; and (iii) competing data traffic.

5.1. Scenario I: links with unequal contact capacities and convergence to high performance

In this scenario, low capacity access links (namely, links AB and AC in Fig. 7) form the network bottleneck; long-haul links (BD and CD) do not restrict data flow; and memory buffers at all nodes are abundant thus never dropping packets. DTTP is compared against SProt-1 and SProt-2. Each of the three protocols is evaluated in a separate simulation run. SProt-1 and SProt-2 arbitrarily select the first and second path, respectively (namely, paths A–B–D and A–C–D), while DTTP probes network performance across both paths, and dynamically adjusts its data route accordingly.

In the respective graphs for this scenario, we present (i) the time required by each of the three protocols to successfully deliver all data (file delivery time) and (ii) the total size of retransmitted packets induced during the file transfer. The file is initially stored at the rover (original sender), and the task is to reliably transmit it to the ground station on Earth (final receiver). We note that "total size of retransmitted packets" involves aggregate retransmissions across the path to destination, so that the total retransmission effort of the protocol in question is reflected. More specifically, we calculate total retransmissions across links AB and BD for SProt-1; total retransmissions across links AC and CD for SProt-2; and total retransmissions across all previous four links (AB, BD, AC, and CD) for DTTP.

Fig. 9a shows file delivery time when the propagation delay between Mars and Earth is 5 min, and the packet error rate present on the long-haul links (BD and CD) equals 1%. The file size varies from 10 to 50 Mbytes. For a 10-Mbyte file: SProt-1 needs around 12.7 h to deliver the file at the ground station on Earth; SProt-2 requires 7.7 h; and DTTP completes the file transfer after 9.4 h. For file size equal to 20 Mbytes, the respective delivery times are approximately: 28.4 h for SProt-1; 15.4 h for SProt-2; and 15.6 h for DTTP. Also, a 50-Mbyte file is delivered after: 68.7 h by SProt-1; 35.7 h by SProt-2; and 37.4 h by DTTP. The respective retransmission effort of all three protocols is shown in Fig. 9b. More specifically, during the 10-Mbyte file transfer, total size of retransmissions is about the same for each of SProt-1, SProt-2, and DTTP: approximately 0.11 Mbytes. Similarly, each of the three protocols induces approximately the same amount of retransmissions when transmitting a 20-Mbyte file (about 0.23 Mbytes of retransmitted data), as well as in the case of a 50-Mbyte file (size of retransmissions is 0.56 Mbytes).

File delivery time and size of retransmissions when the propagation delay between Mars and Earth is 10 min and packet error rate is 1%, is shown in Figs. 9c and d, respectively. Similarly, Figs. 9e and f refer to one-way delay of 20 min and packet error rate of 1%. Similar results to those presented in Figs. 9a–f are shown in Figs. 10a–f, but the latter group of figures refers to packet error rate of 5%.

Observing the related graphs (i.e., Figs. 9a and b; 9c and d; 9e and f; 10a and b; 10c and d; and 10e and f), reveals the performance gains of DTTP. In all cases, DTTP achieves fast data transfers, while at the same time it retains size of retransmissions at low levels. However, blind and fixed path selection might either coincidentally achieve fast data delivery (as is the case for SProt-2), or fail to efficiently exploit alternate routing paths (hence, the increased file delivery times for SProt-1). DTTP outperforms SProt-1: DTTP achieves up to 47% reduction in data receipt time (Fig. 10e), resulting in data transfers that complete 35 h sooner than

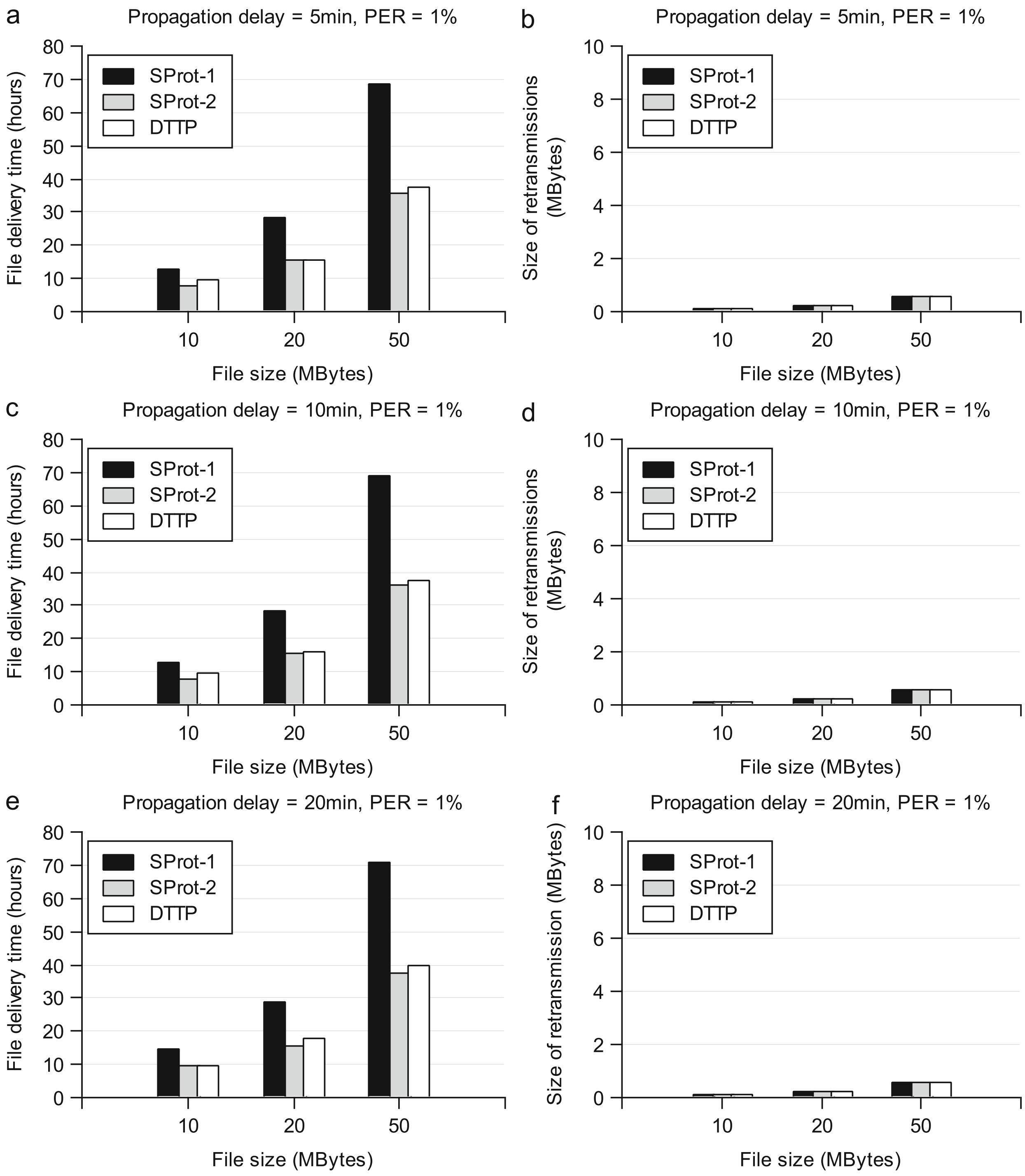

Fig. 9. Scenario I. File delivery time and size of retransmissions for protocols SProt-1, SProt-2, and DTTP. (a) File delivery time; (b) size of retransmissions; (c) file delivery time; (d) size of retransmissions; (e) file delivery time; and (f) size of retransmissions.

SProt-1. It is also noteworthy that as the file to be transmitted increases in size (which translates to longer operation of DTTP), then DTTP's performance converges to and ultimately equates to that of the accidentally well-performing SProt-2 protocol.

5.2. *Scenario II: higher communication contact capacity does not guarantee better performance*

In this network setting, the rover's access links (links AB and AC in Fig. 7) indirectly form the network bottleneck,

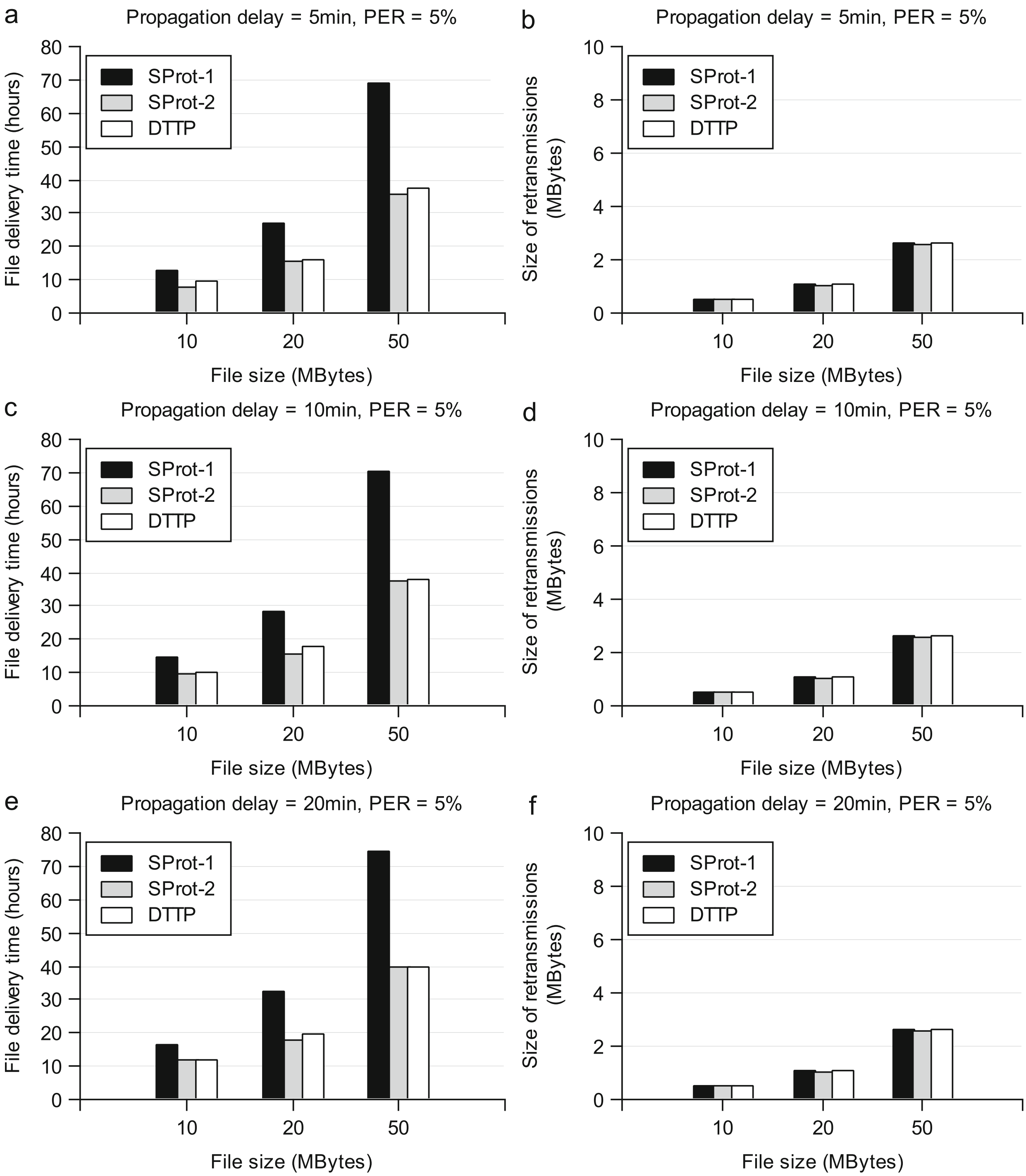

Fig. 10. Scenario I. File delivery time and size of retransmissions for protocols SProt-1, SProt-2, and DTTP. (a) File delivery time; (b) size of retransmissions; (c) file delivery time; (d) size of retransmissions; (e) file delivery time; and (f) size of retransmissions.

and effective data rate is largely affected by storage capacity at the relay satellites (nodes B and C). More specifically, storage capacity of node B is set to 3000 packets, and that of node C is set to 2000 packets. Link AC offers higher bandwidth than link AB. Since contact duration for both connections is the same (see connectivity schedule in Fig. 8), then the resultant contact capacity[3] of AC-connection is higher than that of AB-connection. In other words, during one transmission round, the rover can inject more

[3] *ContactCapacity = Bandwidth*ContactDuration.*

data into the network when forwarding its packets to relay satellite C instead of relay satellite B. Nevertheless, as verified in the results of this scenario, emitting more data into the network does not necessarily mean that all that data can be serviced by the network. Excessive data rates can lead to storage buffer depletion, and cause packet drops.

DTTP periodically measures goodput achieved when forwarding data through either of the two available paths (A–B–D and A–C–D), and correctly selects to route its data mainly via the first path. Indeed, though the first path offers contacts of lower capacity than the second path, it achieves better performance, since storage capacity on the nodes across the first path does not confine data flow. On the contrary, the intermediate node (namely, relay satellite C) on the second path cannot service all incoming packets, even if the associated contact capacity is higher than that of the first path. As a result, goodput performance through path A–C–D is degraded and retransmissions increase in size.

It is remarkable that DTTP completes file delivery sooner than SProt-1 and SProt-2 in all cases without exception (see "File delivery time" in Figs. 11a, c, e, 12a, c, and e). Compared to SProt-1, DTTP decreases file delivery time up to 17% (for 20-Mbyte file, Fig. 12a), thus completing file transfer about 3.6 h sooner than SProt-1. Similarly, when DTTP is compared against SProt-2, DTTP achieves decrease in file delivery time up to 34% (file size is 10 Mbytes, Fig. 12c), and successfully delivers all data 4.6 h earlier than SProt-2. Moreover, DTTP keeps retransmission effort low and comparable to size of retransmissions induced by SProt-1 protocol, whereas SProt-2 induces relatively high levels of retransmissions (see "Size of retransmissions" in Figs. 11b, d, f, 12b, d, and f). Maximum retransmission size for each of the three protocols occurs when the file size is 5 Mbytes, propagation delay between Mars and Earth is 20 min, and packet error rate on Mars-to-Earth links is set to 5%: 2.7 Mbytes for SProt-1, 28.6 Mbytes for SProt-2, and 3.5 Mbytes for DTTP.

Graphs in Fig. 13 show buffer occupancy at intermediate nodes (i.e., relay satellites B and C in Fig. 7), when propagation delay between Mars and Earth is 5 min, packer error rate for long-haul links is assumed 1%, and size of data for transmission is 10 Mbytes. Figs. 13a and b refer to one simulation run and depict buffer occupancy at both relay satellites, as induced by DTTP running on the network. Fig. 13c pertains to another simulation run where SProt-1 is used. Similarly, Fig. 13d refers to a third simulation where SProt-2 is utilized. First of all, we notice a repetitive fluctuation in buffer occupancy that appears mainly as a result of connectivity schedule (Fig. 8): links are iteratively activated and deactivated. In-network storage resources are almost entirely exploited. On the first path, DTTP uses up to 74% of the relay satellite A's storage (Fig. 13a). DTTP (and any other protocol under the same network conditions, such as SProt-1 in Fig. 13c) cannot fully exploit the satellite's storage capacity, because contact volume (i.e., the product of bandwidth and contact duration) does not suffice to completely fill the satellite's memory. However, DTTP completely utilizes storage capacity of relay satellite B (Fig. 13b), which can accommodate 2000 packets.

5.3. Scenario III: transient competing traffic

This scenario captures the effects of competing traffic in the network. During some interval of the total simulation duration, the lander on Mars (node E, Fig. 7) injects data into the network towards the ground station on Earth (node D). This transient traffic occurs on two consecutive communication contacts between the lander and one relay satellite (node B). Each of these two contacts lasts for the same length of time as any contact between the rover and the relay satellites, and both of the contacts affect transmission rounds Nos. 3 and 4, as they occupy buffer space on-board the relay satellite-B. We examine the implications on the performance of DTTP and SProt protocols, as well as the responsiveness of DTTP to degradation of its performance. For this scenario, storage capacity of the relay satellites (i.e., nodes B and C) is set to 3000 packets.

As seen in Fig. 14a, DTTP and SProt-1 manage to complete file delivery relatively fast. More specifically, both DTTP and SProt-1 need about 16.7 h to deliver a 20-Mbyte file, whereas SProt-2 requires five additional hours. In addition, DTTP outperforms SProt-1 in terms of induced retransmissions: for the case of a 20-Mbyte file transmission, SProt-1 aggregate retransmissions amount to 3.1 Mbytes, while total retransmissions of DTTP equal only 1.7 Mbytes (see Fig. 14b). SProt-2, on the other hand, induces even less retransmissions than DTTP; but we should also note that SProt-2 forwards data exclusively through path A–C–D, which is never traversed or affected by the lander's competing traffic. Similar observations can be made for file sizes of 10 Mbytes and 50 Mbytes (Figs. 14a and b).

Finally, we provide a detailed inspection of DTTP's dynamic routing functionality in Fig. 15. We examine the case of a 20-Mbyte file transfer (Scenario III, Table 1). The lower graph in Fig. 15 depicts which access link (AB or AC)—and which path, by extension—is selected by DTTP on the Mars rover at the beginning of each transmission round. By default, DTTP selects path-AB for the 1st transmission round, and path-AC for the 2nd transmission round: in Section 4.1 we refer to this time period as the *probing phase*. The upper graph in Fig. 15 presents the performance across each of the two paths, as it is perceived locally by DTTP at the rover. Performance across a path essentially expresses the size of data (in Mbytes) that were successfully transmitted (that is, received by the next node and acknowledged) during the previous transmission round. Thus, looking at the upper graph at the beginning of transmission round No. 3, the path-AB curve appears above the path-AC curve (i.e., performance across AB is better than that across AC, goodput-wise). As a result, DTTP selects path-AB to forward data to during the 3rd transmission round (see path selection in lower graph). However, at this point, competing traffic (that originates from the Mars lander) has already entered the network, and utilizes

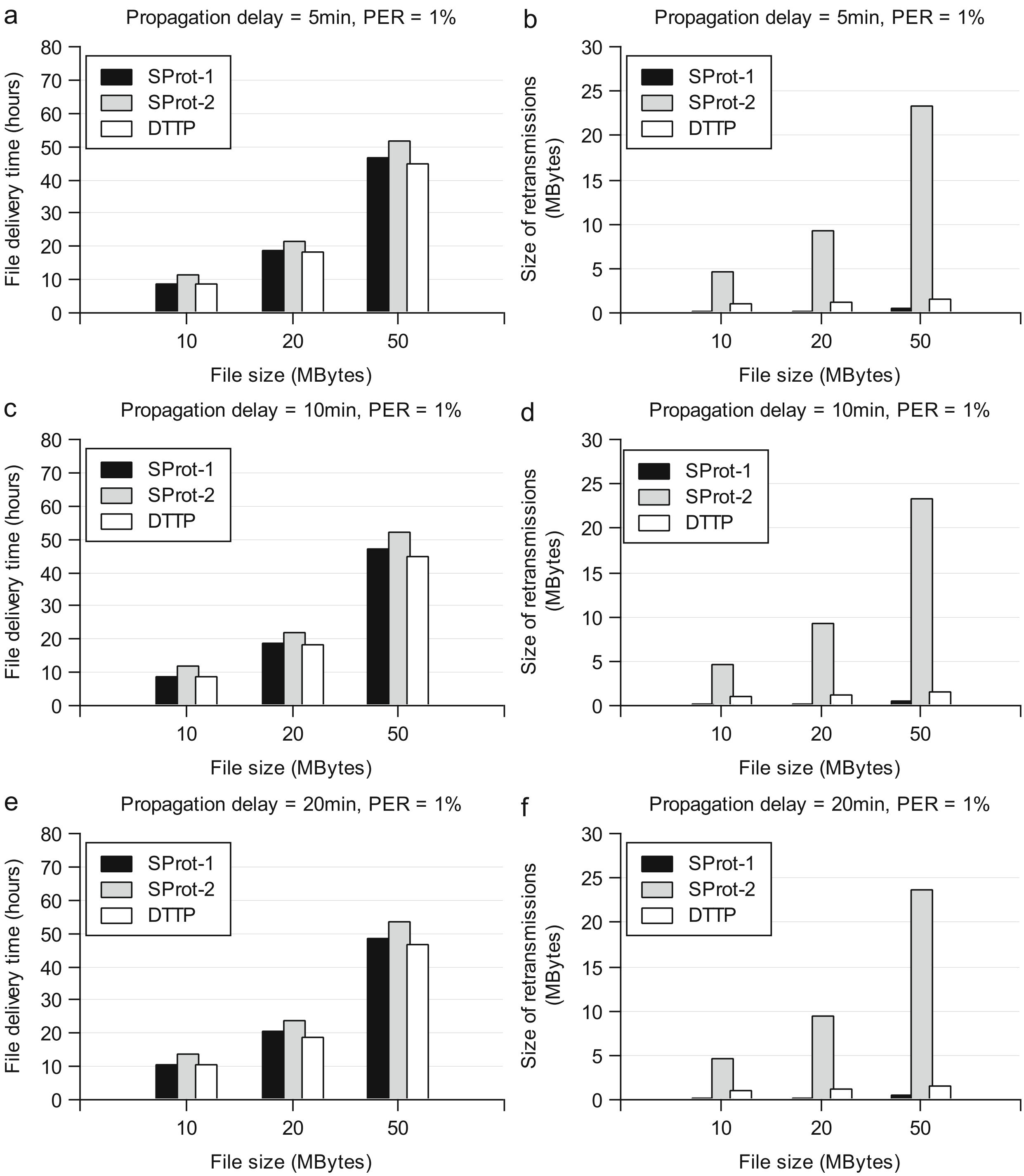

Fig. 11. Scenario II. File delivery time and size of retransmissions for protocols SProt-1, SProt-2, and DTTP. (a) File delivery time; (b) size of retransmissions; (c) file delivery time; (d) size of retransmissions; (e) file delivery time; and (f) size of retransmissions.

network resources on the path E–B–D. The resultant performance degradation for DTTP across path-AB is depicted in the upper graph at the beginning of the fourth transmission round (path-AB curve now appears below path-AC curve). Thus, DTTP once again alters its routing path and selects path-AC at the beginning of the 4th round (see lower graph). Note that the ascending performance of path-AB in the upper graph (transmission rounds Nos. 5 and 6) is due to the fact that our path selection algorithm biases towards paths that have

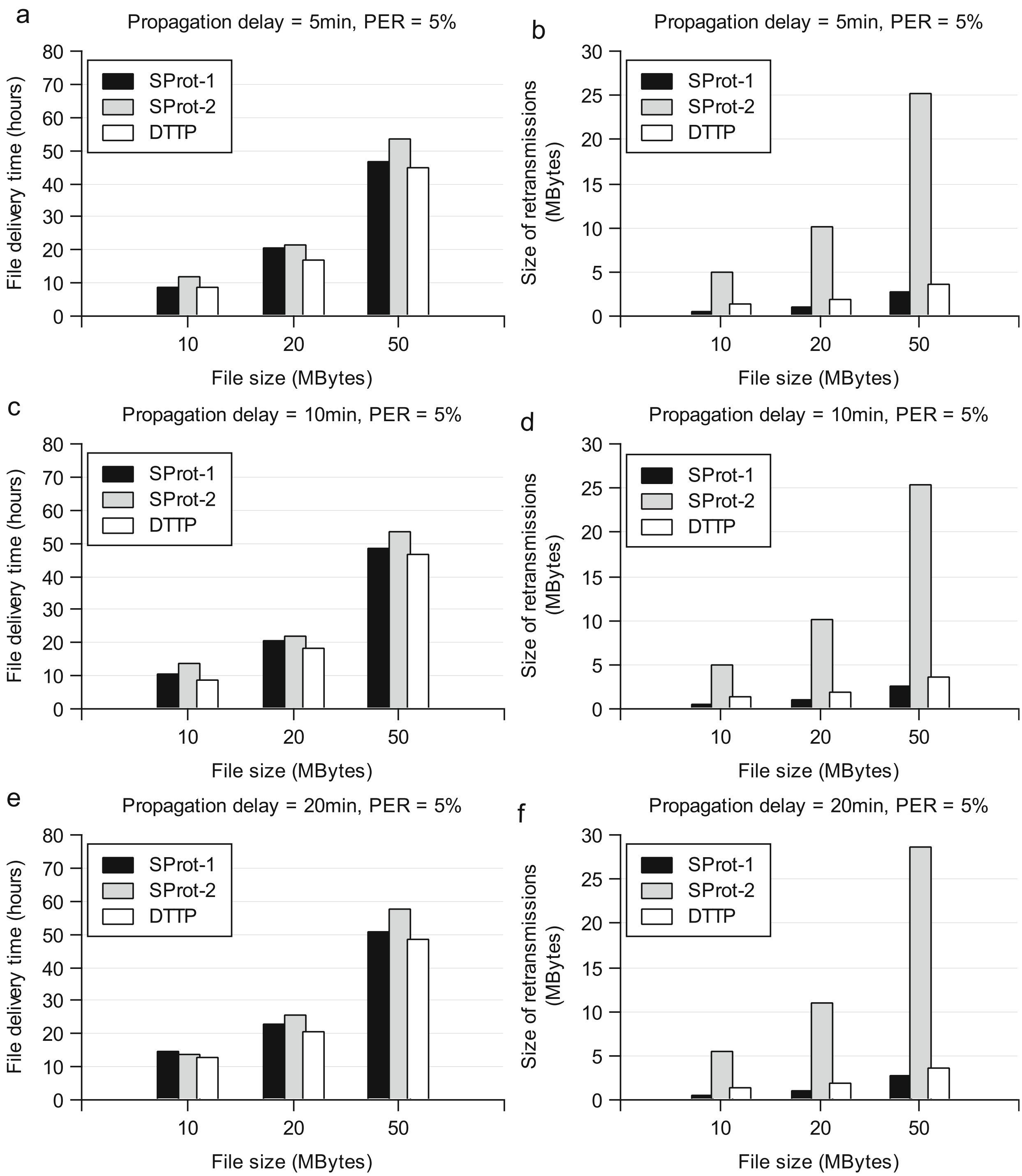

Fig. 12. Scenario II. File delivery time and size of retransmissions for protocols SProt-1, SProt-2, and DTTP. (a) File delivery time; (b) size of retransmissions; (c) file delivery time; (d) size of retransmissions; (e) file delivery time; and (f) size of retransmissions.

recently demonstrated high performance (for more details, see Section 4.1). Therefore, at the beginning of transmission round No. 6, when the transient competing data flow from the lander has ended, DTTP reverts to the high-performing path-AB.

6. Conclusions and future work

Delay-tolerance does not necessarily mean belated space communications. We demonstrate that reliability can be coupled with fast communications. We also show

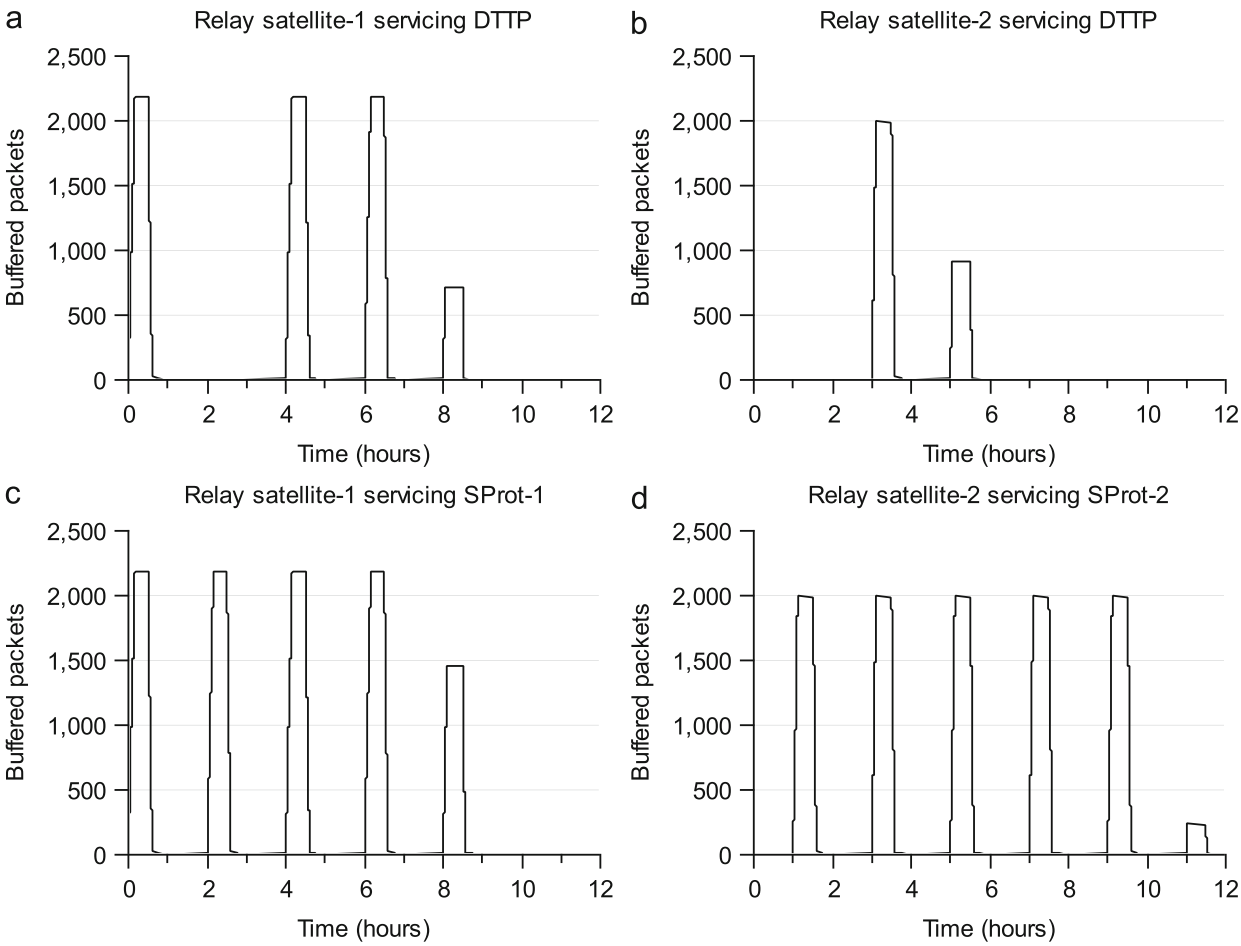

Fig. 13. Scenario II. Buffer occupancy at relay satellites. (a–d) Buffer occupancy.

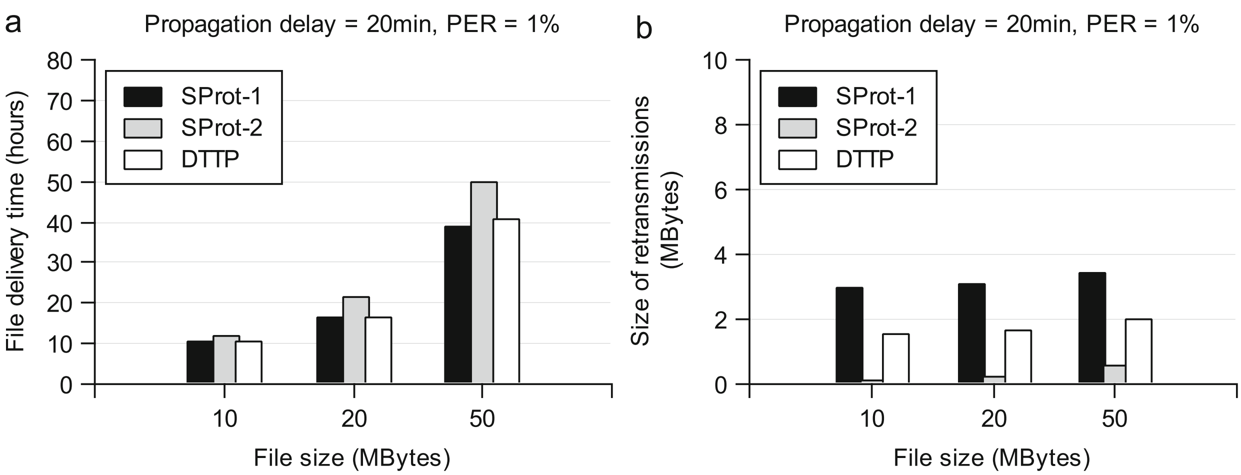

Fig. 14. Scenario III. File delivery time and size of retransmissions for protocols SProt-1, SProt-2, and DTTP. (a) File delivery time and (b) size of retransmissions.

that dynamic route selection in space is possible. Through simulations, we verify that local information about network state can enhance transports over networks with long delays and disconnections. Delay-Tolerant Transport Protocol (DTTP) periodically measures its performance across available paths, and adjusts data

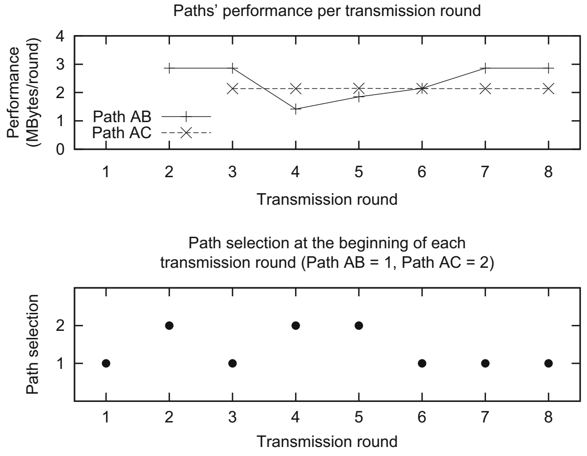

Fig. 15. Scenario III. DTTP: dynamic routing based on paths' performance measurements.

routing accordingly. We show that DTTP reduces time of data delivery at the final destination, and keeps the size of retransmissions low.

In this paper, we have examined scenarios where local measurements enhance ultimate *end-to-end performance*. However, local information alone cannot always second efficient data transfers on Delay-Tolerant Network infrastructures. When network dynamics cannot be timely perceived by some network node (due to connectivity schedule, for example), then the node's ability to adapt its transmission/routing strategy and enhance its performance is greatly degraded. As a future task, we will examine a network messaging system, where space nodes periodically inform each other about network load/performance. We have already implemented a probing mechanism for DTTP that can capture *local network state*. We will investigate whether (and at which degree) sharing knowledge of network state can benefit space data transfers, and lead to even more efficient routing procedures in space networks.

References

[1] The Consultative Committee for Space Data Systems (CCSDS), The Official Web Site: 〈http://www.ccsds.org/〉.

[2] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson, S. Wissler, Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission, SPACOMM 2009, Colmar, France, July 2009.

[3] C. Samaras, I. Komnios, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou, N. Peccia, Extending Internet into Space—ESA DTN testbed implementation and evaluation, in: First International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT 2009), Workshop on Research activities funded by ESA in Greece, Athens, Greece, May 18–20, 2009.

[4] Extending Internet into Space—Phase 2, ESA/ESOC DNT/IP Testbed Implementation and Evaluation, web site: 〈http://www.hai-rt.gr/projects/ExtendingInternetIntoSpace/〉.

[5] C.V. Samaras, V. Tsaoussidis, DTTP: A Delay-Tolerant Transport Protocol for Space Internetworks, Second ERCIM Workshop on eMobility, Tampere, Finland, May 2008.

[6] M. Bajracharya, M.W. Maimone, D. Helmick, Autonomy for Mars Rovers: Past, Present, and Future, Computer, vol. 41(12), 2008.

[7] Preliminary Planning for an International Mars Sample Return Mission, Report of the International Mars Architecture for the Return of Samples (iMARS) Working Group, June 1, 2008.

[8] ESA's ExoMars Mission, web site: 〈http://www.esa.int/SPECIALS/ExoMars/index.html〉.

[9] K. Birnbaum, W.H. Farr, J. Gin, B. Moision, K.J. Quirk, M.W. Wright, Tests of optical communications for deep space show promise, SPIE Newsroom, 18 June 2009.

[10] B. Smutny, et al., 5.6 Gbps optical intersatellite communication link, in: Proceedings of SPIE, vol. 7199, 2009, p. 719906.

[11] Jet Propulsion Laboratory, Strategic Technology Directions 2009, JPL Publication 400-1385.

[12] R. Gladden, et al., Mars relay coordination lessons learned, in: 2005 IEEE Aerospace Conference, March 2005, Big Sky, Montana.

[13] Consultative Committee for Space Data Systems, Space Packet Protocol, Recommendation for Space Data System Standards, CCSDS 133.0-B-1, Blue Book, Issue 1, CCSDS, Washington, DC, September 2003.

[14] V. Cerf, et al., Delay-Tolerant Networking Architecture, RFC 4838, Internet Engineering Task Force, April 2007.

[15] K. Scott, S. Burleigh, Bundle Protocol Specification, RFC 5050, Internet Engineering Task Force, November 2007.

[16] Consultative Committee for Space Data Systems, Rationale, Scenarios, and Requirements for DTN in Space, CCSDS 734.0-G-0, Draft Green Book, Issue 0, CCSDS, Washington, DC, March 2010 Work in progress.

[17] L. Wood, W.M. Eddy, P. Holliday, A bundle of problems, in: IEEE Aerospace Conference, Big Sky, Montana, March 2009.

[18] V. Tsaoussidis, I. Psaras, C.V. Samaras, G. Papastergiou, Some comments on Delay-Tolerant Networking for Space Communications, Technical Report TR-DUTH-EE-2008-4, Democritus University of Thrace, Greece.

[19] G. Papastergiou, I. Psaras, V. Tsaoussidis, Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs, Computer Communications, Special Issue of Computer Communications on Delay and Disruption Tolerant Networking, vol. 32, Issue 16, 15 October 2009, pp. 1757–1767.

[20] Consultative Committee for Space Data Systems, CCSDS File Delivery Protocol (CFDP), Recommendation for Space Data System Standards, CCSDS 727.0-B-4, Blue Book, Issue 4, CCSDS, Washington, DC, January 2007.

[21] M. Ramadas, S. Burleigh, S. Farrell, Licklider Transmission Protocol—Specification, RFC 5326, Internet Engineering Task Force, September 2008.

[22] S.C. Burleigh, Dynamic Routing for Delay-Tolerant Networking in Space Flight Operations, SpaceOps 2008, Heidelberg, Germany, May 2008.

[23] W. Ivancic, W.M. Eddy, D. Stewart, L. Wood, J. Northam C. Jackson, Experience with Delay-Tolerant Networking from orbit, preprint for the International Journal of Satellite Communications and Networking, special issue for best papers of ASMS 2008.

[24] Interagency Operations Advisory Group (IOAG), web site: 〈http://www.ioag.org/〉.

[25] Recommendations on a Strategy for Space Internetworking, Report of the Interagency Operations Advisory Group, Space Internetworking Strategy Group, July 28, 2008.

[26] Consultative Committee for Space Data Systems, Mars Mission Protocol Profiles—Purpose and Rationale, Informational Report, CCSDS 740.0-G-1, Green Book, Issue 1, CCSDS, Washington, DC, July 2008.

[27] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, R. Patra, Implementing Delay Tolerant Networking, Technical Report IRB-TR-04-020, Intel Research Berkeley, December 2004.

[28] The Network Simulator ns-2, web site: 〈http://www.isi.edu/nsnam/ns/〉.

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.