INTERNATIONAL JOURNAL OF SATELLITE COMMUNICATIONS AND NETWORKING
Int. J. Satell. Commun. Network. 2014; **32**:127–140
Published online 23 October 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sat.1043

Mars to Earth communications through orbiters: Delay-Tolerant/ Disruption-Tolerant Networking performance analysis

C. Caini[1], R. Firrincieli[1], T. de Cola[2,*,†], I. Bisio[3], M. Cello[4] and G. Acar[5]

[1]*DEIS–Department of Electronics, Computer Science and Systems, University of Bologna, Bologna, Italy*
[2]*DLR–German Aerospace Center, Institute of Communications and Navigation, Weßling, Germany*
[3]*DITEN–Department of Electrical, Electronic, and Telecommunication Engineering and Naval Architecture, University of Genoa, Genoa, Italy*
[4]*ECE–Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA*
[5]*ESA–European Space Agency, Keplerlaan 1-2200 AG, Noordwijk ZH, The Netherlands*

SUMMARY

Delay-Tolerant/Disruption-Tolerant Networking (DTN) architecture will be used in future deep space missions, to enable autonomous networking operations and disruption-tolerant data communications. Therefore, it is worth analyzing the performance of the DTN Bundle Protocol (BP) in a realistic deep space environment, reproducing the characteristics of Mars missions. After a comprehensive introduction on data communications between Mars and Earth, the paper presents the essential features of both the BP DTN architecture and the Licklider Transmission Protocol (LTP), adopted here as BP convergence layer on deep space links, thanks to its ability to cope with the very long delays typical of this environment. The focus of our experiments is on analysis of the bundle flow from a Mars lander to an Earth control center through an intermediate relay node, for which two configurations are considered, inspired to Odyssey and Mars Reconnaissance Orbiter missions, respectively. Results are obtained by means of a test bed consisting of some GNU/Linux personal computers running either Interplanetary Overlay Network (ION) or DTN2 BP implementations. The analysis of results aims to highlight the role played by BP and LTP in tackling the challenges of Mars to Earth communications. Copyright © 2013 John Wiley & Sons, Ltd.

Received 23 January 2012; Revised 28 May 2013; Accepted 24 June 2013

KEY WORDS: Deep space networking; DTN; Bundle Protocol; Licklider Transmission Protocol; CCSDS

1. INTRODUCTION

Increasing interest in space exploration and scientific experiments in the solar system has resulted in an ever-growing number of space missions. Besides, advances in space technologies, such as more effective transmission techniques employed over either radio frequency or optical media, greater on-board computation capabilities and improved networking features, paved the way for the deployment of more complex space networks. This resulted in the planning of manned missions in space, thus requiring the support of typical Internet-like services, which must be enabled along with the more common telemetry and telecommand applications. As a consequence, a set of requirements for these new services has been worked out by space agencies, as listed in Table I.

In addition, the consolidation of the Delay-Tolerant/Disruption-Tolerant Networking (DTN) architecture [1] opened the door to autonomous networking operations in space. The concept of DTN was first introduced in the InterPlanetary Networking Special Interest Group (IPNSIG) and then refined in the Internet Research Task Force (IRTF) DTN Research Group. The consequent standardization has resulted in a series of Request for Comments (RFCs) related to the DTN architecture [2], the

*Correspondence to: T. de Cola, DLR–German Aerospace Center, Institute of Communications and Navigation, Weßling, Germany.
†E-mail: tomaso.decola@dlr.de

Table I. Link requirements for new services in space [15].

	User	Channel content	Latency	Number of channels	Bit rate per channel	Total bit rate
Operational	Base	Speech	NRT	2	10 kbps	20 kbps
		Engineering	NRT	1	100 kbps	100 kbps
	Astronauts	Speech	NRT	4	10 kbps	40 kbps
		Helmet camera	NRT	4	100 kbps	400 kbps
		Engineering	NRT	4	20 kbps	80 kbps
	Human transports	Video	NRT	2	1.5 Mbps	3 Mbps
		Engineering	NRT	2	20 kbps	40 kbps
	Robotic rovers	Video	NRT	8	1.5 Mbps	12 Mbps
		Engineering	NRT	8	20 kbps	160 kbps
	Science orbiters	Quick look	NRT	4	1 Mbps	4 Mbps
		Engineering	NRT	4	20 kbps	80 kbps
High rate	Human transports	HDTV (medical, PIO)	NRT	2	20 Mbps	40 Mbps
		Hyperspectral imaging	1 day	1	150 Mbps	150 Mbps
	base	HDTV	1 day	1	20 Mbps	20 Mbps
	Robotic rovers	Surface radar	1 day	1	100 Mbps	100 Mbps
		Hyperspectral imaging	1 day	1	150 Mbps	150 Mbps
	Science orbiters	Orbiting radar	1 day	2	100 Mbps	200 Mbps
		Hyperspectral imaging	1 day	2	150 Mbps	300 Mbps
Total	Design reference mission (DRM) and HDTV-operational					80 Mbps
	Add robotic operations and DRM hyperspectral imaging					480 Mbps
	Add science orbiters					980 Mbps

NRT, non real-time; HDTV, high definition television; DRM, design reference mission.

Bundle Protocol (BP) specification [3], the Licklider Transmission Protocol (LTP) (in terms of motivation [4], protocol specification [5] and security extensions [6]), the Bundle Security Protocol [7], and others. More recently, Consultative Committee for Space Data Systems (CCSDS) has also started standardization with a DTN working group in the Space Internetworking Services area. Its aim is to elaborate DTN internetworking requirements for future deep space scenarios [8], and new specifications of BP [9] and LTP [10] protocols tailored to the operating conditions of space missions. In particular, the building blocks of the overall DTN architecture in space are inspired to the Space Interworking Strategy Group report [11].

The scientific community has already dedicated attention to the performance assessment of DTN in deep space networks. Preliminary studies on Mars to Earth DTN communications are presented in [12] and results of Extrasolar Planet Observation and Deep Impact Extended Investigation (EPOXI)/Deep Impact Network Experiment (DINET) I and DINET II preliminary DTN tests in [13].

The aim of this paper is to analyze recent Mars to Earth space missions in the light of DTN concepts. As shown in the paper, the availability of orbiters, acting as intermediate nodes between Mars and Earth, is mainly dictated by their advantages in terms of link budgets. However, they fit particularly well with the store-and-forward BP transmission mechanism, which relies on permanent storage at intermediate nodes to cope with intermittent links and network partitioning. The idea of applying DTN to Mars to Earth communications through orbiters is not new. In fact, the idea and some preliminary results, obtained by simulations, have already been presented in [12]. This paper, however, is much more focused on the experimental evaluation of BP and LTP performance. The main aim is to investigate the role played by some distinctive features of BP and LTP protocols in tackling the challenges of the Mars to Earth scenario. For this purpose, the analysis of bundle 'status reports' provided by the BP proved to be essential, as it allowed an in-depth study of the bundle flow from a Mars lander to an Earth control center. Tests were carried out by means of GNU/Linux personal computers running both DTN2 [14] and Interplanetary Overlay Network (ION) [15]. BP implementations, a secondary aim was the practical assessment of potential (and limits) of the two most common BP implementations currently available, especially about their interoperability.

The remainder of the paper is organized as follows: in Section 2 we introduce the Mars to Earth communication scenario investigated in the rest of the paper; Section 3 describes the DTN BP architecture; Section 4 gives an overview of LTP; Section 5 describes the test bed used; Section 6 is devoted to the analysis of results; finally, conclusions are drawn in Section 7.

2. MARS TO EARTH COMMUNICATIONS SCENARIO

2.1. Overview

The scenario considered in this paper stems from recent exploration missions to Mars (Figure 1). From the communications point of view, a Mars lander can communicate (e.g., send images) to Earth via two alternative relays, represented by two Mars orbiters. On Earth, data are received by two ground stations (not represented in the figure for simplicity), which forward them to two control centers, through wired connections.

The availability of relay nodes is essential for high throughput figures, resulting in a larger volume of data forwarded to the destination, with respect to Direct-to-Earth links usually employed in the past. In particular, the unavailability of relay nodes drastically reduces the downlink data rate at a few kbps, because of the very little transmission power available on the planetary nodes. In addition, the time intervals where both landers and Earth are in sight (or 'contact windows') are very limited, thus reducing the data volume that can be transferred. Both the problems are solved by the deployment of relay nodes (typically orbiters). The downlink data rate in the first leg (from the lander to the orbiter) is increased to about 128 kbps, thanks to the shorter propagation path, and in the second leg (from the orbiter to the Earth) from 1 Mbps to 100 Mbps, thanks to the higher transmission power. Moreover, the possibility to store data on the orbiter releases the constraints of direct visibility from the lander to the Earth, as data can be first sent from the lander to the orbiter, and then sent to the Earth as soon as the next orbiter-to-Earth contact window becomes available. As it will be shown later, the DTN BP store-and-forward mechanism fits perfectly well in this scenario.

An important aspect is represented by the storage capacity of orbiters. Usually, space relay nodes are employed for scientific experiments and memory is limited to a few tens of Megabytes. Relaying operations, however, would demand for dedicated large mass storages, and future space missions should envision the use of larger memory units on orbiters to take full advantage of the store-and-forwarding mechanism.

2.2. Reference scenario

The Mars to Earth scenario considered is inspired by the case of the Phoenix lander exploration mission [16] to Mars, with slight modifications to allow a broader investigation. The Phoenix mission started in 2007 and one aim was to make use of a relay network already available in

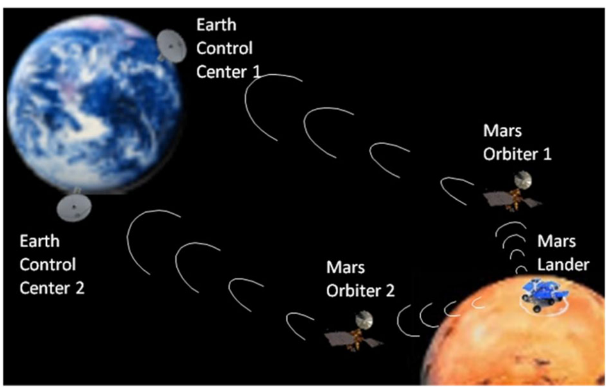

Figure 1. Mars to Earth scenario.

DOI: 10.1002/sat

order to improve the transmission efficiency achieved in the previous Spirit and Opportunity missions. These latter missions used two landers, which could directly communicate to Earth via a direct X-band link, thus suffering from the low performance intrinsic in direct communications, just discussed. As remarked earlier in the text, the availability of a relay network greatly improves the performance, especially in terms of data volume download [17]. Concerning the relay network, the characteristics of our two orbiters are inspired by the following real spacecrafts:

- Mars Odyssey [18]. This started in 2001 and was the first Mars relay satellite conforming to the CCSDS Proximity-1 Space Link Protocol [19]. The orbiter is set on a circular orbit at an average altitude of 400 km. The relay payload supports both a forward link (command) and a return link (telemetry). Data rates of 8, 32, 128, and 256 kbps are possible, with optional (7,1/2) convolutional coding.
- Mars Reconnaissance Orbiter (MRO) [20]. Launched on August 12, 2005 it is set on a circular orbit at an average altitude of 300 km and carries a new generation of relay radio: the Electra Proximity Payload, conforming to the CCSDS Proximity-1 Space Link Protocol and supporting data rates from 1 kbps to 1 Mbps, with (7,1/2) convolutional coding. MRO is actually being used for relaying operations over Mars.

In our scenario, mention should be made of the duration of contact windows. Those between the lander and the orbiters vary in time but are usually limited to a few minutes. We assumed, as average values, 12 min and 8 min for Odyssey and for MRO, respectively. Contact windows between the orbiter and the Earth are much longer: we assumed, as average values, 11 h and 8 h for Odyssey and for MRO, respectively. It is also important to bear in mind that lander-orbiters and orbiters-ground station contact windows are not aligned, thus requiring store-and-forward transmissions.

The propagation delay in the links between orbiters and landers (~1 ms) has been assumed negligible with respect to that exhibited by the links between orbiters and Earth ground stations, which is much larger and variable, depending on the Earth and Mars variable distance. The minimum distance (~0.36 astronomical unit (AU). AU, corresponding to Earth-Sun average distance estimated to 149 598 106 km) was considered in our tests for the communication between MRO and Earth, corresponding to a one-way delay of 180 s. By contrast, the average distance (~0.72 AU) was considered for the Odyssey case, corresponding to a double link delay (360 s). Concerning link data rates, for the orbiter-lander link, an uplink of 128 kbps and a downlink of 4 kbps are assumed for both orbiters. On the contrary, the orbiter-Earth data rate is different in the two cases. Odyssey-Earth link has a downlink of 110 kbps and an uplink of 1 kbps. MRO-Earth link has a data rate of 4 Mbps on the downlink and 1 kbps on the uplink. The indicated data rate figures refer to the values actually configured on transceivers in the space missions, although slightly higher values could have been used as well. Overall network link characteristics are summarized in Table II.

As far as the overall network protocol stack is concerned, it is assumed that all nodes implement a full CCSDS-DTN protocol architecture. The BP is implemented over the LTP in all nodes.

Table II. Network link characteristics.

	Link	Data rate [kbit/s]	Delay [s]
Case 1 (Odyssey)	Lander-Odyssey	128	~0
	Odyssey-Lander	4	
	Odyssey-Earth Control Center	110	360
	Earth Control Center-Odyssey	1	
Case 2 (MRO)	Lander-MRO	128	~0
	MRO-Lander	4	
	MRO-Earth Control Center	4000	180
	Earth Control Center-MRO	1	

MRO, Mars Reconnaissance Orbiter

DOI: 10.1002/sat

3. BUNDLE PROTOCOL DELAY-TOLERANT/DISRUPTION-TOLERANT NETWORKING ARCHITECTURE

3.1. Bundle protocol and convergence layer adapters

In order to support communication in challenged environments, the BP DTN architecture [2, 3] is based on the insertion of a new 'Bundle layer', between application and lower layers (usually transport). The related protocol (BP) can interface with lower layers through 'Convergence Layer Adapters', (CLAs), as shown in Figure 2. Various CLAs have been defined, including those for Transmission Control Protocol (TCP) [21], User Datagram Protocol (UDP) [22], and LTP [23]. In this new architecture, transport protocol end-to-end features are confined to homogeneous network segments (namely, A, B, and C in Figure 2), whereas end-to-end data transfer across the heterogeneous network is provided by the bundle layer; large data packets called 'bundles' are exchanged between DTN nodes through a store-and-forward relay.

The main innovations of DTN architecture are summarized in the succeeding texts.

3.1.1. DTN overlay. By installing the BP on end-points and some intermediate nodes, the end-to-end path is divided into multiple DTN hops. On each hop, a different protocol stack can be used or when the same stack is retained, as it commonly happens, just different protocols (e.g., TCP and UDP) or different versions of the same protocol (e.g., TCP variants). The advantage is that each DTN node on a path can use whatever convergence layer (CL) is best suited for the next forwarding operation, which is essential to cope with the varying characteristics of heterogeneous network links (e.g., very long delay deep space links).

3.1.2. Storage at intermediate nodes. To cope with long delays and possible lack of connectivity, information (i.e., data bundles) must be stored at intermediate DTN nodes for long periods and when the custody option [2], [24] is enabled, only on persistent memory (e.g., local hard disks). The custody option is one of the most innovative features of DTN and deserves to be illustrated in further detail. When enabled, DTN nodes on the path are asked to take 'custody' of a bundle [22]. When a node accepts it, it becomes its new 'custodian', which essentially means that it takes responsibility for any retransmissions. In this way, the bundle can be retransmitted by the current custodian instead of the sender. This reduces latency and frees the memory resources of the sender, which no longer needs to keep a copy of the bundle. The custody option increases reliability and is particularly useful whenever the sender has limited memory and/or power resources. Last but not least, the custody option also makes the DTN architecture resilient against temporary node failures, similar to blackouts, reboots, or even hardware faults (except of course the loss of the persistent memory support).

3.1.3. Bundle fragmentation. An interesting feature of DTN BP is the possibility of fragmenting bundles. It can work either a priori ('proactive fragmentation'), to meet constraints on the maximum amount of data transferable ('contact volume') on a DTN hop at each availability time ('contact time'),

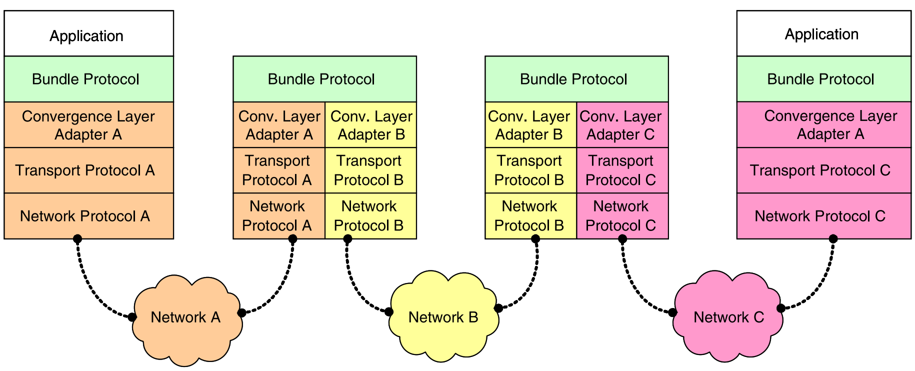

Figure 2. Delay-Tolerant/Disruption-Tolerant Networking architecture and protocol stack.

DOI: 10.1002/sat

or a posteriori ('reactive fragmentation'), to avoid retransmitting already acknowledged data in the presence of disruptive channels. Note, however, that this feature is not implemented in ION, because it is assumed that in a controlled space environment bundle dimension can be set in accordance to contact volume constraints (i.e., no larger than contact volumes).

3.1.4. Bundle status reports. Another distinctive feature of DTN BP is status reports, sent by DTN nodes to report the transmission 'status' of a bundle. Each report refers to one specific bundle and is sent to the DTN node specified in the 'report_to:' field of the bundle header, which may or may not coincide with bundle source. If it does, the source is informed of the bundle progress, much similar to courier parcel tracking. If it does not coincide, supervision is performed by a different node, which can be useful in challenging situations (e.g., when the sender has limited resources, when its link to the network is disrupted, or short lived, or mono directional, etc.). The bundle status reports [2] are the following:

- Bundle reception—when a bundle arrives at a DTN node.
- Custody acceptance—when a node has accepted custody of a bundle.
- Bundle forwarded—when a forwarded bundle departs from a DTN node.
- Bundle deletion—when a bundle is discarded.
- Bundle delivery—when a bundle is delivered to an application at the destination node.
- Acknowledged by application—when a bundle has been processed by an application at the destination node. This generally involves specific action on the receiving application's part.

With few exceptions, the transmission of status reports is optional. They must be explicitly requested by the source node by setting a specific bit on the bundle header, although a DTN node is generally not obliged to accept this request, (e.g., to save bandwidth).

3.2. Bundle protocol implementations

3.2.1. Delay-Tolerant/Disruption-Tolerant Networking 2 Bundle Protocol reference implementation. The DTN2 [14] is designed as an experimental platform to test the BP features on a real implementation, which could also be used for real world deployment. In addition to the BP, the DTN2 package also contains some DTN basic applications (dtnping, dtnsend, etc.) and the DTNperf_2 tool for DTN performance evaluation [25] used in our experiments. To speed up DTN2 installation, a 'packet' version of DTN2 has been developed by the authors for Ubuntu and Debian GNU/Linux distributions. This packet enables a faster installation, easier maintenance, and improved consistency of the software (all DTN2 installations are the same).

3.2.2. Interplanetary Overlay Network: National Aeronautics and Space Administration Bundle Protocol implementation. The ION is a BP implementation by National Aeronautics and Space Administration Jet Propulsion Laboratory, with contributions from Ohio and other universities [15]. Although compliant with Request for Comments on DTN architecture and BP, as DTN2, it is explicitly focused on deep space applications [15]. As in these environments, TCP cannot be used because of lengthy link delays, ION also contains an implementation of LTP. ION also includes the Contact Graph Routing, a routing algorithm specifically designed for scheduled intermittent connectivity, typical of space links, where actual link availability is known in advance, being because of the motion of spacecrafts and planets. ION contains also a full implementation of the Bundle Security Protocol, the BP security extensions. Among the supported convergence layer adapters we have: TCPCL (interoperable with DTN2), UDPCL (likewise interoperable with DTN2), and LTPCL. ION offers some features of particular interest here, similar to scheduled links, which have not yet been implemented in DTN2. It should be noted, however, that in ION, scheduled links require the use of LTP at convergence layer. Moreover, as mentioned earlier in the texts, ION does not implement reactive bundle fragmentation, unlike DTN2. The latest ION code is available as open source from Sourceforge [26]. It has also been recently included in Debian and Ubuntu.

DOI: 10.1002/sat

4. LICKLIDER TRANSMISSION PROTOCOL

4.1. Motivation

The LTP was designed to provide retransmission-based reliability over links with extremely long round trip time (RTT) and/or frequent interruptions in connectivity. Hence it is appropriate as convergence layer in DTN architectures [4, 5]. Its peculiar characteristic is its minimum 'chattiness', making it suitable for deep space point-to-point links. In fact, there is no connection set-up, thus saving some RTTs in the initial phase. Besides, no real congestion or flow control involving exchange of information between nodes is implemented in the protocol. Particularly important is the concept of session: for each new LTP block [10] (found in the succeeding texts below), a new session, that is, a new LTP thread, is initiated. The maximum number of possible parallel sessions poses a requirement of corresponding storage capacity at a node. This acts as a sort of flow control, although no information is exchanged between the two corresponding LTP nodes, and configuration of maximum number of parallel sessions is done statically, for example, during space mission planning phase. Finally, LTP may also implement a rate-based congestion control, which avoids saturating the network node buffers, based on transmission link configuration, which is notified to each LTP node by means of a periodically updated contact table. In the case of ION, for each link between two nodes, this table contains the propagation delay, the available rate, and the contact durations, which are also used by LTP to schedule data transmission. In this way, two LTP nodes can send and receive data as soon as (and as long as) the link between them is available, which leads to optimal utilization of the contact window.

4.2. Protocol description

Processing of data units from the Bundle Layer to the underlying layer is done in three steps: bundle aggregation, block segmentation, and transmission. First, the BP LTP convergence layer adapter combines the bundles from the BP layer into one LTP block [10], which is in turn forwarded to LTP protocol entity. The number of bundles being aggregated to form a LTP block depends on the maximum LTP block size. Only entire bundles can be inserted in an LTP block; therefore, a block may consist of only one bundle. Block size is selected according to space mission requirements and may be set as any multiple of bundle size. This feature has been specifically introduced for deep space communications where scarce bandwidth calls for minimum protocol overhead. Next, the LTP protocol entity divides any incoming block into a number of LTP segments whose maximum size depends on the underlying layer protocol to which they are finally forwarded.

The LTP segments are assigned two levels of reliability: red and green. The former requires segments to be reliably delivered to destination, using NAK-based automatic repeat request (ARQ) mechanisms. By contrast, the latter requires no reliability, and therefore no retransmission mechanism is present. A block contains both a red and a green part or only one of the two. The way reliability levels are assigned to blocks or part of blocks is implementation dependent, although appropriate mapping from the extended classes of service [9] could be used. It is worth noting that mapping procedures are not yet available in ION implementations, where all blocks are treated as red.

Data transfer between two LTP peers also includes the exchange of administrative reports from destination to sender to solicit the retransmission of the missing red segments or to notify correct block or segment reception. Checkpoint (CP) segments can be issued from the sender asking the receiver to send a report segment to acknowledge all received segments, or alternatively to inform the sender about the missing segments (i.e., NAK). In both cases, the sender will eventually generate a report acknowledgment, followed by retransmitted segments in case of losses. Finally, the sender can signal the completion of red part (end of red part) or of the entire block (end of block) through a dedicated flag (also marked CP) in the last segment of the red section, or of the block, respectively.

4.3. An example of LTP session

A typical LTP session is shown in Figure 3. An LTP block, made up of six segments, is transmitted over a deep space link: all the segments except 3 and 4 are received correctly. On reception of the last

DOI: 10.1002/sat

segment (both end of red part and end of block, flagged as CP), a report indicating the segments received (#1, #2 and #5, #6 in the figure) is sent by receiver to sender. In turn, the sender generates an acknowledgment for this report and then retransmits the two missing segments flagging the second #4 as CP. This is acknowledged by a report to signal full reception of the LTP segment, in turn confirmed by a new acknowledgment. The LTP session can now be closed at both ends and the corresponding buffer space freed.

5. TEST BED DESCRIPTION

In order to reproduce the scenario in Figure 1, an ad-hoc test bed based on GNU/Linux machines was set up. The layout is shown in Figure 4.

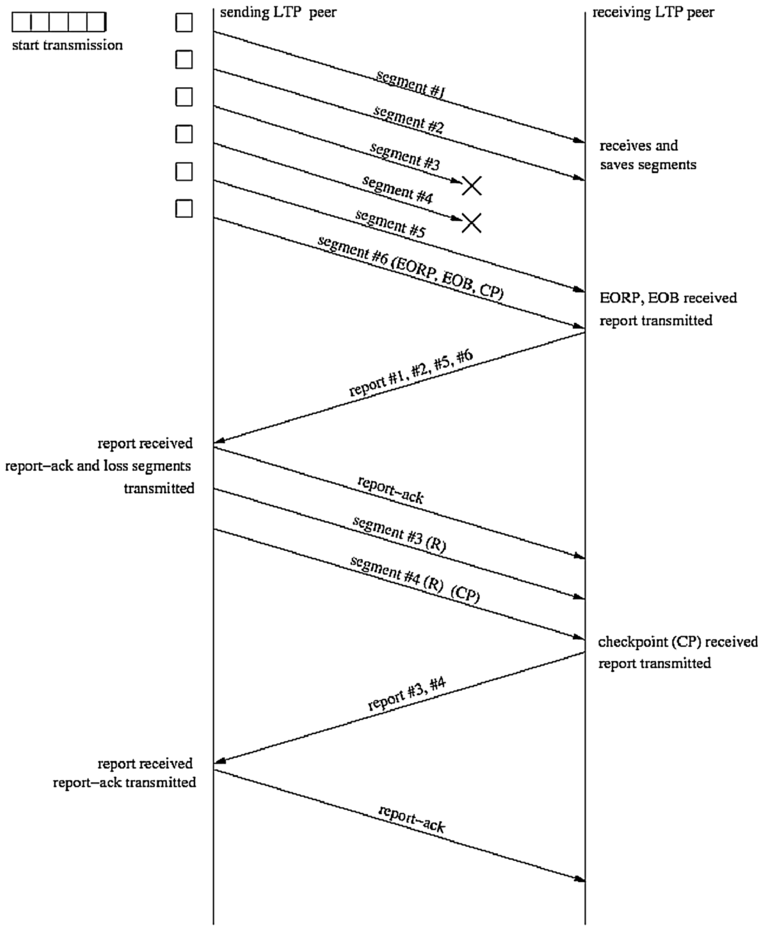

Figure 3. An example of Licklider Transmission Protocol segment transmission flow.

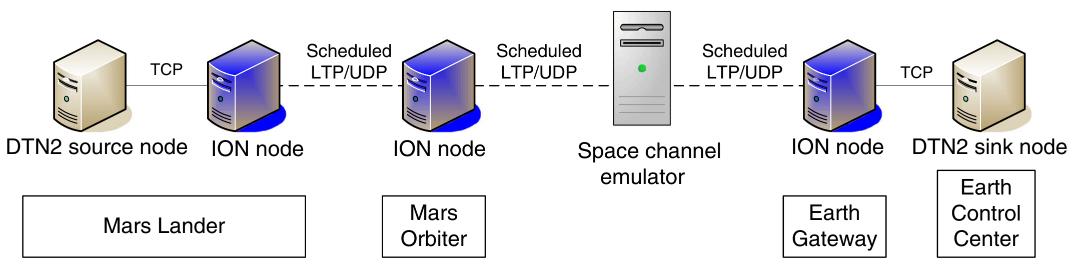

Figure 4. Layout of the deep space testbed.

DOI: 10.1002/sat

In the test bed, both DTN2 and ION nodes are present. DTN2 is installed on the two test bed end-nodes, in order to allow the use of the DTNperf_2 tool for generating the data bundle flow with the desired characteristics (in particular size and number of bundles). Moreover, DTNPerf_2 allows the user to track the transmission flow by collecting at sender side the bundle ‘status reports’ generated by all DTN nodes, a feature of great interest for the analysis of DTN traffic. The ION suite is particularly valuable because it includes an embedded implementation of the LTP, which is especially suitable in deep space links. Moreover, ION, together with LTP, offers the possibility of utilizing scheduled links and fine-tuning of the transmission rate, other two features of great interest in deep space communications. For these reasons ION is installed on the three internal DTN nodes of our test bed, at the borders of the two space communications links (lander-orbiter and orbiter Earth Gateway), where LTP is used as a convergence layer. In the DTN2-ION links, where the almost negligible delays cannot affect the performance, TCP is used as a convergence layer. It is worth remarking that one LTP segment is encapsulated in many UDP datagrams (LTP/UDP tag in Figure 4), bundles are encapsulated in one TCP segment. In turn both UDP/TCP datagrams/segments are processed by Internet protocol (IP) and then by Ethernet protocol entities.

In addition to the five DTN nodes mentioned, the test bed also includes a channel emulator, based on the NistNet tool [27], to generate delay and packet losses in the deep space links, between the orbiters and the Earth stations.

6. NUMERICAL RESULTS

Tests were performed using DTNperf_2 client and server on the two test bed end-points. Bundle size is set to 50 kB, the maximum LTP block to 60 kB, the maximum number of LTP sessions to 128, the maximum LTP segment (to be encapsulated in a UDP segment) to 1 kB. Lander to orbiter and orbiter to Earth Control Center links use LTP CLA and are scheduled. The data rates of these LTP links are set as shown in Table II, whereas contact time durations are summarized in Table III. In the tests, for convenience, the actual start times of these contact windows were set to reduce total test duration, without impacting on performance. In particular, we cut to 120 s the interval between the first lander to orbiter and the orbiter to Earth Control Center contact windows, which usually lasts up to several hours.

6.1. Case 1: Odyssey orbiter

In this case, bundle transfer is initiated at the Mars lander directed to the Earth Control Center, through the Odyssey orbiter. Bundle logs, BP ‘status reports’ collected by DTNperf_2, are shown in Figure 5 and Figure 6, for a lossless (PER = 0) and a lossy (PER = 3%) deep space link, respectively.

6.1.1. Lossless deep space link. At time 0, the DTNperf_2 client is started on the DTN2 source on Mars lander: ten bundles are immediately created, passed to the BP running on this machine, sent to the second DTN node (the ION machine on Mars lander), and taken into custody (‘Custody on Lander’ status reports,). When the lander to orbiter link becomes available (at 120 s), the bundles are progressively transmitted to the third DTN node, that is, the Mars Orbiter, where they are taken into custody (‘Custody on Orbiter’ series). The next DTN hop is between the orbiter and the Earth Gateway, through the Mars to Earth link, which in this case has been assumed loss free and with a one-way delay of 360 s. This link becomes available only at 960 s, that is, after the closure of the lander to orbiter link. Note that this would have prevented the use of ordinary network

Table III. Contact time durations.

	Contact	Duration [s]
Case 1	Lander-Odyssey	720
	Odyssey-Earth control center	39600
Case 2	Lander-MRO	480
	MRO-Earth control center	28800

MRO, Mars Reconnaissance Orbiter

DOI: 10.1002/sat

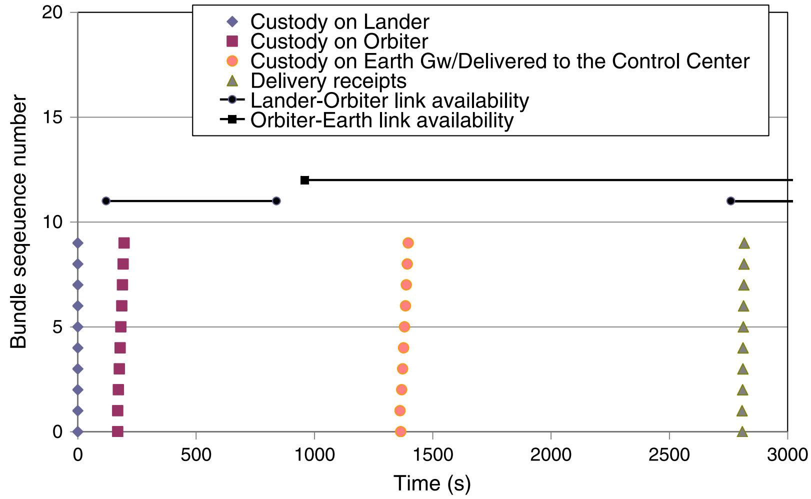

Figure 5. Case 1 (Odyssey) bundle reports logs: PER = 0 on the deep space link.

Figure 6. Case 1 (Odyssey) bundle reports logs: PER = 3% on the deep space link.

architecture, because of the lack of a continuous end-to-end path between the sender and destination, whereas BP DTN store-and-forward can easily cope with it. All ten bundles are then received and taken into custody by the Earth gateway after a little more than 360 s (one half RTT) from link availability ('Custody on Earth GW/Delivered to the control center' series). This excellent result is because of the absence of losses in the deep pace link and to the chat-free characteristics of the LTP protocol. In more details, each 50 kB bundle is encapsulated into one LTP block, which in turn is segmented into 50 1 kB UDP segments, which are all directly transmitted to the Earth Gateway. In the absence of losses, all the UDP segments are correctly received and reassembled into one LTP block; this is de-encapsulated, and its payload is passed to the BP as the original bundle. Note that, although the LTP destination has to send one or more 'reception report' to the LTP source for each LTP block to confirm actual reception of segments, this procedure does not delay the block reassembly in the absence of losses. This explains why the bundles are received after a little more than just one half RTT, that is, basically the mere propagation time, which is the absolute minimum for the transmission delay.

Let us return to the bundle flow. The double label in the 'Custody on Earth GW/Delivered to the control center' series is justified by the fact that bundle transfer in the last DTN hop, between the Earth gateway and the Earth control center, where bundles are finally delivered to the DTNperf_2 server (running DTN2), is almost instantaneous. Actually, this series is obtained from the timestamp field

DOI: 10.1002/sat

of status delivered reports. Although the bundle transfer from Mars to Earth is now complete, bundles receipts still have to be sent back to the bundle source on Mars. This is accomplished in two steps. In the first, delivered status reports are transferred to the orbiter and taken into custody. In the second, when the second lander to orbiter contact window starts (at 2760 s), bundle receipts are finally delivered to the source ('Delivery receipts' series).

6.1.2. Lossy deep space link. Here, the same experiment is repeated, but introducing a high loss (PER = 3%) on the Mars to Earth link. In this case (Figure 6), some bundles (namely 0, 3, 4, and 6) are received on Earth after approximately half RTT from the start of the Mars to Earth contact window, as before ('Custody on Earth GW/Delivered to the Control Center' series). The other bundles, however, need about 1.5 RTT. This is explained by considering that each bundle is transferred through an LTP block, which in turn requires about 50 UDP segments (each consisting of 1 kB), leading to 1.5 segment losses per block on average. These losses prevent the immediate reassembly of the LTP block. Their recovery by LTP takes about 1 additional RTT (half to transmit the 'reception report' with the partial ACK of the block, another half to retransmit the missing segments). Of course, it may happen that some blocks do not suffer any losses (0, 3, 4, and 6 in Figure 6), resulting in immediate delivery to the Earth Gateway after a half RTT, as before.

6.2. Case 2: Mars Reconnaissance Orbiter

Here, we assume that the bundle transfer is carried out using the MRO orbiter. The same tests as before were carried out and lossless results are shown in Figure 7 and lossy in Figure 8.

6.2.1. Lossless deep space link. For Figure 7, the general comments already given for the corresponding Odyssey case (Figure 5) still hold. The numerical values, however, are different, mainly because of the shorter one-way delay assumed between the Orbiter and the Earth (i.e., 180 s), but also, though less significantly, because of different contact windows and link speeds.

As soon as the lander to orbiter link becomes available (at 120 s), bundles generated on the Mars lander ('Custody on Lander' series) are transferred to the orbiter where they are temporarily taken into custody ('Custody on Orbiter'). Then, when the contact window of the orbiter to Earth link starts, they are transmitted to the Earth Gateway and delivered to the control center ('Custody on Earth GW/Delivered to the control center'). The orbiter to Earth hop requires about 180 s, that is, only one half RTT, as in the absence of losses no retransmissions are required. The correct reception of bundles is acknowledged by delivered status reports, which have to go back to the source on the lander ('delivery receipts' series).

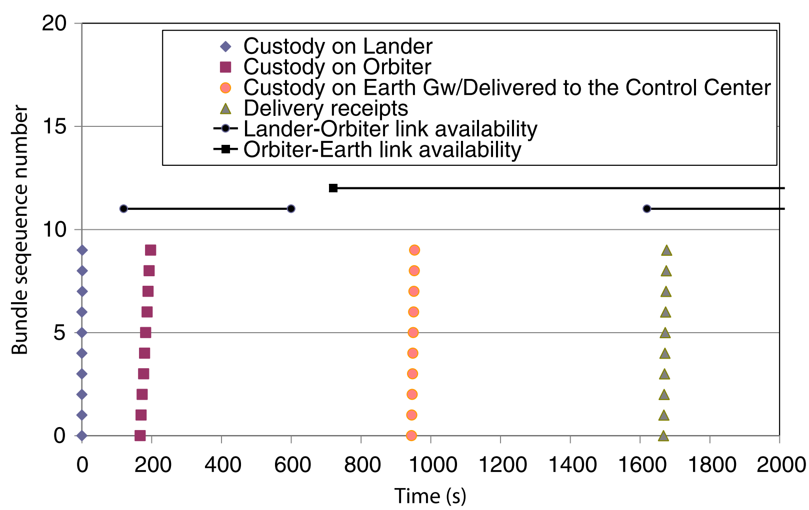

Figure 7. Case 2 (Mars Reconnaissance Orbiter) bundle reports logs: PER = 0 on the deep space link.

DOI: 10.1002/sat

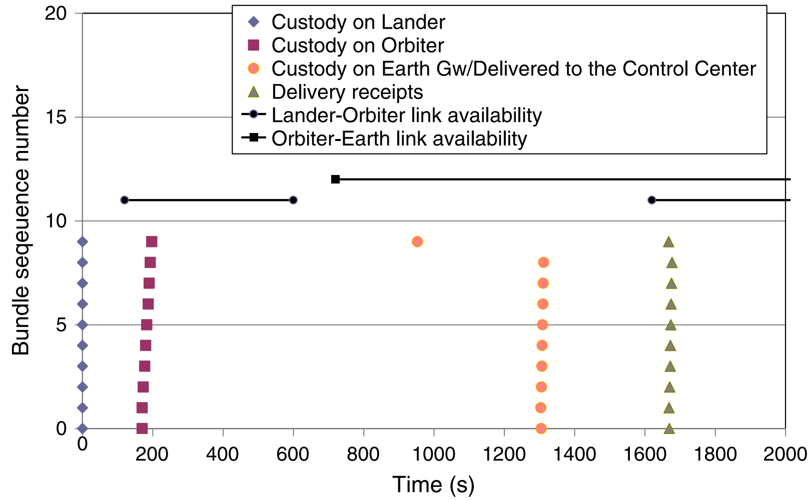

Figure 8. Case 2 (Mars Reconnaissance Orbiter) bundle report logs: PER = 3% in the deep space link.

6.2.2. Lossy deep space link. By introducing a high loss percentage (PER = 3%) on the deep space link (Figure 8), we have the same behavior of the corresponding Odyssey case (Figure 6). Losses need to be recovered by LTP, which takes about 1 additional RTT (360 s here). However, by chance one bundle (#9) is not affected by any loss and is delivered in only one half RTT. Note that this leads, as in the corresponding Orbiter case, to disordered bundle delivery.

7. CONCLUSIONS

The paper addressed the performance of DTN architecture over Mars to Earth scenarios, modeled on the case of Phoenix missions, with Odyssey and MRO orbiters as relay nodes. Two DTN implementations, DTN2 and ION, were used to evaluate the performance of BP and LTP protocols in a dedicated test bed. Performance assessment focused on the reliable transmission of bundles from a lander to its Earth control center.

Both lossless and lossy deep space links were considered. In the former, as bundles are not affected by LTP segment losses, they were correctly received at destination after only about half RTT. This is an excellent result, as it corresponds to the propagation time, that is, the lower bound for bundle delivery. In the latter, as a number of LTP segments were lost, an additional RTT was required to allow retransmission. This is an equally good result, as it is the minimum additional delay for a reliable protocol based on retransmission. This fully confirms the suitability of the LTP protocol as BP convergence layer in deep space environments.

A further enhancement of LTP capabilities in presence of link errors is represented by the introduction of erasure codes implemented between LTP and underlying layers, in order to avoid retransmission loops that would further increase the delivery latency. This aspect currently under standardization within CCSDS [28] is a further point of investigation and is considered for future extensions of this work, resulting in the real implementation of erasure codes into ION software package.

ACKNOWLEDGMENTS

The work was funded by the European Space Agency (ESA/ESTEC) in the framework of the SatNEx III (Satellite Network of Experts) project, as part of CoO1-Task 1 ‘Applicability of Delay and Disruption Tolerant Networks to Space Systems’.

REFERENCES

1. Burleigh S, *et al*. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Commun Mag* 2003; **41**(6):128–136.
2. Cerf V, *et al*. Delay-tolerant networking architecture, *IETF RFC 4838*, 2007.
3. Scott K, Burleigh S. Bundle protocol specification, *IETF RFC 5050*, 2007.
4. Ramadas M, Burleigh S, Farrell S. Licklider transmission protocol—motivation, *IETF Request for Comments RFC 5326*, 2008.
5. Ramadas M, Burleigh S, Farrell S. Licklider transmission protocol—specification, *IETF Request for Comments RFC 5326*, 2008.

DOI: 10.1002/sat

6. Ramadas M, Burleigh S, Farrell S. Licklider transmission protocol—security extensions, *IETF Request for Comments RFC 5327*, 2008.
7. Symington S, Farrell S, Weiss H, Lovell P. Bundle security protocol specification, *Internet RFC 6257*, 2011.
8. Rationale, scenarios, and requirements for DTN in space. Rationale, scenarios, and requirements for DTN in space. *CCSDS 734.0-G-1. Green Book. Issue 1*. Washington, DC, USA: CCSDS, 2010.
9. CCSDS bundle protocol specification. *CCSDS 734.2-R-1. Red Book. Issue 1*. Washington, DC, USA: CCSDS, 2012.
10. Licklider transmission protocol (LTP) for CCSDS. *CCSDS 734.1-R-2. Red Book. Issue 2*. Washington, DC, USA: CCSDS, 2012.
11. Recommendations on a strategy for space internetworking. *Space Internetworking Strategy Group report*, 2009.
12. Krupiarz CJ, Jennings EH, Pang JN, Schoolcraft JB, Seguí JB, Torgerson JL, Spacecraft data and relay management using delay tolerant networking. *AIAA 9th International Conference on Spacecraft Operations (SpaceOps)*, Rome, Italy, 2006; 19–24.
13. Wyatt J, Burleigh SC, Jones R, Torgerson L, Wissler S. Disruption tolerant networking flight validation experiment on NASA's EPOXI mission. *SPACOMM, First International Conference on Advances in Satellite and Space Communications*. 2009, Colmar, France, 20-25 2009; 187–196.
14. DTN2 Bundle protocol reference implementation code: (Available from: http://sourceforge.net/projects/dtn/).
15. Burleigh S. Interplanetary overlay network (ION) an implementation of the DTN bundle protocol. In *the Proc. of 4th IEEE Consumer Communications and Networking Conference*, 2007; 222–226.
16. Lewicki CA, Krajewski J, Ilott P. Phoenix mars scout UHF relay-only operations. *AIAA 9th International Conference on Spacecraft Operations (SpaceOps)*, Rome, Italy, 2006; 19–24.
17. Edwards CD, Arnold B, DePaula R, Kazz G, Lee C, Noreen G. Relay communications strategies for Mars exploration through 2020. Elsevier, *Acta Astronaut* 2006; **59**:310–318.
18. Makovsky A, Barbieri A, Tung R. Odyssey telecommunications. *JPL, Descanso Series on Design & Performance Summary*, 2002.
19. Proximity-1 space link protocol–data link layer. *Recommended Standard, CCSDS 211.0-B-4. Blue Book. Issue 4.* Washington, DC, USA: CCSDS, 2006.
20. Taylor J, Lee DK, Shambayati S. Mars reconnaissance orbiter telecommunications. *JPL, Descanso Series on Design & Performance Summary*, 2006.
21. Demmer M, Ott J. Delay tolerant networking TCP convegence layer protocol. Internet-Draft, 2008; work-in-progress (Available from: http://tools.ietf.org/html/draft-irtf-dtnrg-tcp-clayer) Accessed 2010-09-28.
22. Kruse H. UDP convergence layers for the DTN bundle and LTP protocols. Internet-Draft, 2008. work-in-progress, (Available from: http://tools.ietf.org/html/draft-irtf-dtnrg-udp-clayer) Accessed 2010-09-28.
23. Burleigh S. Delay tolerant networking LTP convergence layer (LTPCL) adapter. Internet-Draft, 2010. work-in-progress, (Available from: http://tools.ietf.org/html/draft-burleigh-dtnrg-ltpcl) Accessed 2010-09-28.
24. Fall K, Hong W, Madden S. Custody transfer for reliable delivery in delay tolerant networks. *Technical Report IRB-TR-03-030,* Intel Research, Berkeley, 2003; 1–6. Available at DTNRG web site.
25. Caini C, Cornice P, Firrincieli R, Livini M. DTNperf 2: a performance evaluation tool for delay/disruption tolerant networking. In *Proc. of E-DTN =2009*, St.-Petersburg, Russia, 2009; 1–8.
26. ION reference code: (Available from: http://sourceforge.net/projects/ion-dtn/).
27. Carson M, Santay D. NIST Net: a Linux-based network emulation tool. *ACM SIGCOMM Comput Commun Rev* 2003; **33**(3):111–126.
28. de Cola T, Paolini E, Liva G, Calzolari G. Reliability options for data communications in the future deep-space missions. *Proc IEEE* 2011; **99**(11):2056–2074.

AUTHORS' BIOGRAPHIES

Carlo Caini received the Dr. -Ing. degree (summa cum laude) in Electronics Engineering from the University of Bologna, Italy, in 1986. Since 1990, he has been with the Department of Electronics Computer Science and Systems of the same University, where at present, he is an associate professor of Telecommunications. His main scientific interests are in the field of satellite communications, with a special emphasis on the study (and design) of network protocols and architectures for satellite and wireless applications. He participates to several international research projects, and he is a member of the Satellite Network of Experts co-funded by the European Space Agency. He is an author of many international publications on these and other topics. He is a member of the IEEE Communications Society.

Rosario Firrincieli received his Master's and PhD degrees in Telecommunications Engineering from the University of Bologna, Italy, in 2001 and 2006, respectively. From 2006 to 2011, he was a senior researcher at ARCES, University of Bologna, Italy, where he worked on enhanced transport protocols, PEP solutions, DTN architectures over wireless network, congestion control algorithms for unicast and multicast protocols, traffic shaping, retransmission techniques, and UL-FEC coding techniques. During this period, he also spent 10 months as a visiting researcher at the Department of Computer Science, Henry Samueli School of Engineering and at the Applied Sciences, UCLA, USA. In 2007, he co-founded meXage S.r.l., a software consulting company where he currently works as senior consultant. Dr. Firrincieli is a co-author of many scientific publications on International Journals and Conferences, and he was a co-recipient of the Best Paper Award at SPACOMM'09 and IEEE IWSSC'09 conferences with his work on DTN.

DOI: 10.1002/sat

Tomaso de Cola was born in Manosque, France, on April 28, 1977. He received the 'Laurea' degree (with honors) in Telecommunication Engineering in 2001, the qualification degree as a professional engineer in 2002, and his PhD degree in Electronics and Computer Engineering, Robotics, and Telecommunications in 2010 from the University of Genoa, Italy. From 2002 to 2007, he has worked with the Italian Consortium of Telecommunications (CNIT), University of Genoa Research Unit, as a scientist researcher. Since 2008, he has been with the German Aerospace Centre (DLR), where he is involved in different European projects focusing on different aspects of digital video broadcasting standards, Consultative Committee for Space Data Systems protocols, and testbed design. He is a co-author of more than 30 papers, including international conferences and journals. His main research activity concerns: Transmission Control Protocol/Internet protocol suite, satellite networks, transport protocols for wireless links, interplanetary networks, and delay tolerant networks. Dr. de Cola served on the Technical Program Committee at several Institute of Electrical and Electronics Engineers (IEEE) International Conferences. He is also a member of the IEEE Communications Society.

Igor Bisio was born in Novi Ligure (Alessandria), Italy in 1978. He obtained his 'Laurea' degree at the University of Genoa, Italy in 2002. He obtained his PhD degree in 'Information and Communication Sciences' at the University of Genoa in 2006. His PhD was funded by the Italian Consortium of Telecommunications (CNIT). He is currently an assistant professor and a member of the research staff of the Telecommunication Research Group and, in particular, of the Digital Signal Processing and Satellite Communications and Networking Laboratories at the University of Genoa. Since 2012, he is also the Chair of the IEEE ComSoc Satellite and Space Communications Technical Committee. His main research activity concerns: resource allocation and management for satellite communication systems, digital signal processing over smartphone platforms, and context-awareness.

Marco Cello (member, IEEE) was born in Savona, Italy, in 1983. He received the 'Laurea Magistrale' degree (cum laude) in Telecommunications Engineering in 2008 and his PhD degree in 2012, both from the University of Genoa. Now, he is working at the Electrical and Computer Engineering Department, Polytechnic Institute of New York University. His main research activity concerns: call admission control in data center networks and cloud systems, call admission control in QoS networks, and routing and congestion control in delay tolerant networks.

Guray Acar was born in Antakya, Turkey, in 1974. He completed his BS studies in Electronics Engineering at the Middle East Technical University, Ankara. He holds an MSc degree in Communications and Signal Processing and a PhD degree in Broadband Satellite Networking, both from the Imperial College London, UK. He has been employed at the Purdue University USA, the University of Newcastle, and the London South Bank University, prior to his employment at the Centre for Communications Systems Research, University of Surrey where he participated in several European Space Agency and European projects in satellite communications and networking. Since 2010, he has been at European Space Agency/European Space Research and Technology Centre as a telecommunication system engineer. He provides technical support and participates in R&D activities at MAC, network, and transport layers in satellite communications.

DOI: 10.1002/sat