

Multicolor Licklider Transmission Protocol: An LTP Version for Future Interplanetary Links

ANDREA BISACCHI 

CARLO CAINI , Member, IEEE
University of Bologna, Bologna, Italy

TOMASO DE COLA , Member, IEEE
German Aerospace Center (DLR), Weßling, Germany

The Licklider Transport Protocol (LTP) is the “convergence layer” of choice in Interplanetary networks based on Delay-/Disruption-Tolerant architecture. It was designed for long-delay scheduled-intermittent links, offering either a reliable or an unreliable service, with “red” and “green” parts, respectively. The aim of this article is to present multicolor LTP, an LTP version consisting in a series of enhancements of which the most significant are the use of monochrome sessions, the introduction of an additional orange color offering a “notified” service, and the definition of default link colors. After a thorough examination of basic LTP mechanisms for all color variants, this article discusses two scenarios where orange seems particularly appealing: video streaming and optical interplanetary links. Numerical results offer further insight into the complex LTP mechanisms and also highlight the difference between LTP retransmissions and bundle protocol retransmissions, the latter benefitting from routing reprocessing. Multicolor LTP has already been implemented as an interplanetary overlay network (ION) plug-in and its enhancements have been proposed to Consultative Committee for Space Data Systems Space Internetworking Services Delay-/Disruption-Tolerant Networking working group for a possible inclusion in the next version of LTP specifications (LTPv2).

Manuscript received 27 November 2021; revised 21 March 2022; released for publication 16 May 2022. Date of publication 23 May 2022; date of current version 11 October 2022.

DOI. No. 10.1109/TAES.2022.3176847

Refereeing of this contribution was handled by M. Marchese.

Authors' addresses: Andrea Bisacchi and Carlo Caini are with the Department of Electrical, Electronic and Information Engineering, Advanced Research Center on Electronic System, University of Bologna, 40126 Bologna, Italy, E-mail: (andrea.bisacchi5@studio.unibo.it; carlo.caini@unibo.it); Tomaso de Cola is with the Institute of Communications and Navigation, German Aerospace Center (DLR), 51147 Weßling, Germany, E-mail: (tomaso.decola@dlr.de). (Corresponding author: Carlo Caini.)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

I. INTRODUCTION

Interplanetary networks are characterized by long delays, intermittent scheduled connectivity, asymmetry of links, and possibly relatively high packet loss rates due to variable channel conditions [1]. To cope with these challenges, in the envisioned interplanetary internet ordinary transmission control protocol/internet protocol (TCP)/(IP) architecture should be replaced by Delay-/Disruption-Tolerant Networking (DTN) architecture [2], [3], which is based on the introduction of a new layer, the bundle layer (between application and usually transport) and on its homonymous protocol the bundle protocol (BP) [4]–[6]. In this framework, the protocol below the BP is usually called “convergence layer”.

The BP is present not only on source and destination nodes, but also on a few intermediate nodes, so the end-to-end path is divided into a number of consecutive DTN hops. DTN architecture [7] has two pillars: first, the end-to-end role of transport is now limited to one DTN hop, somewhat extending the TCP splitting performance enhancing proxies architecture; second, bundles, i.e., the BP “packets,” usually of large dimension, can be stored for long periods at all DTN nodes. The former pillar enables the use of different transport protocols on different hops, to match the channel characteristics of each hop; the latter is essential for coping with link intermittency due to the motion of planets and spacecraft. As an example, if we considered the download of images from a rover on mars to a mission control centre (MCC) via a terrestrial gateway, we could imagine having three DTN hops; one, on mars, between rover and lander, a second, interplanetary, between Lander and Gateway station, the last, terrestrial, between gateway station and mission control centre. On both Martian and terrestrial hops the ordinary TCP/IP protocols could be used, as there are no particular challenges. By contrast, on the space hop we would have all the mentioned challenges, therefore, a transport protocol specifically designed for space links, the Licklider Transmission Protocol (LTP), would be necessary. LTP was first standardized by the Internet Research Task Force in [8] and [9], and then by the Consultative Committee for Space Data Systems (CCSDS) in [10]. The original LTP version was intended to offer both reliable and unreliable services, with “red” and “green” parts, respectively.

The aim of this article is to present an enhanced version, multicolor LTP, the result of over 15-year collaboration between the University of Bologna and DLR. During this collaboration we considered LTP both alone [11] and joined with packet layer forward error correcting codes (PL-FEC); the latter concept was first presented in [12], then discussed and further elaborated by the authors and other researchers in [13]–[20]. Multicolor LTP introduces a dozen new features, the most important of which are the compulsory use of monochrome blocks (consisting either of the red or the green part, but not both), the introduction of orange and the possibility of setting a default color to links.

These and others innovations are presented in full detail in this article, starting with a comprehensive explanation

of LTP mechanisms in the different colors. Then, a few application scenarios will be discussed, including BP video streaming, communications on long-delay and large bandwidth delay product (BDP) links, and communications on paths including unidirectional links. A suitable testing scenario will offer the opportunity to present a detailed analysis of LTP mechanisms and their interaction with both routing and PL-FEC.

The innovations described here are to be presented to the Space Internetworking Services (SIS) DTN working group of CCSDS, as a possible contribution to the new LTPv2 standard. Finally, multicolor LTP has already been implemented as a plug-in of interplanetary overlay network (ION) (the NASA-JPL suite of DTN protocols) [21], [22], in a package called Unibo-LTP, recently released as free software.

II. LTP PROTOCOL

A. LTP Main Features

LTP is the convergence layer of choice on interplanetary links, as it minimizes interaction (“chattiness”) between transmitting and receiving engines [8]–[10]. LTP can run on top of either UDP (in test beds, as in this article) or CCSDS space protocols (in real deployments). On top of LTP we will assume BP, but other protocols could be used as well. An interface between BP and LTP may be present, essentially to perform bundle aggregation, but will not be considered here.

Let us list the key features that differentiate LTP from TCP, most aiming to minimize chattiness.

- 1) No connection-establishment phase (by contrast to TCP three-way handshake).
- 2) Rate based transmission speed (not based on feedback).
- 3) Unidirectional data flow (the reverse channel is used only for signaling), to cope with possible channel asymmetry.
- 4) Bundles passed by BP are encapsulated in LTP “blocks”, to be transmitted by independent LTP “sessions” running in parallel to fill the BDP.
- 5) An LTP block is split into a number of LTP “segments,” each passed to UDP, or other CCSDS space protocol.
- 6) Unlike with TCP, LTP acknowledgments (“report segments”) are triggered only by data segments flagged as “checkpoints,” usually at block end.

B. LTP Monochrome Blocks

Years of experience in using standard LTP have shown that LTP blocks could advantageously consist of a single color, either red or green, instead of red, and green parts possibly coexisting in the same block. First, implementation complexity could be reduced; then with monochrome blocks the session, i.e., the process of transferring a block, would naturally assume the color of the block. As colors can be considered as different QoS, there would be clear one-to-one mapping between the QoS required to transfer a

bundle and the session color. For example, if the bundle had to be transferred reliably, a red session should start and the bundle should be inserted into a red block. Conversely, if the unreliable service were required, the new session should be green.

This concept can easily be generalized, by denoting any different service provided by LTP with a different color. At this point, it seems advantageous to introduce at least one additional color, orange, associated to a “notified” service. To summarize, for multicolor LTP, we propose the following color-QoS mapping.

- 1) Red (red sessions transferring red blocks), offering a reliable service.
- 2) Green (green sessions transferring green blocks), offering an unreliable service
- 3) Orange (orange sessions transferring orange blocks), offering the new “notified” service.

Other colors could possibly be added in the future. Note that having the same protocol providing different QoSs seems preferable to having different protocols, each providing a sole QoS. This because the upper and the lower protocol interfaces would be the same, thus simplifying implementation, and also because it would be simpler to select the most appropriate QoS for each bundle.

The QoS choice (the LTP color) should consider both bundle preferences (e.g., its “ECOS” flags [23]) and local link characteristics. For example, a bundle could be transferred as green either because flagged as “unreliable” by the source, or simply because the local link is unidirectional and does not allow any other kind of service. Other examples are possible and will be presented after a closer examination of the various colors.

III. LTP COLORS

A. Red Sessions (Reliable Service)

Let us start with red sessions; the differences between multicolor and standard LTP are few and will be pointed out when necessary.

1) *No Losses*: First, we examine the case of no losses (see Fig. 1); when the transmission session (“export session” in ION) starts, all the segments of the LTP block are sent, the last being flagged as end of red-part (EORP), end of block (EoB), and Checkpoint (CP) [9]. Note that in multicolor LTP there would be no need of the EORP flag (in monochrome sessions the red-part must coincide with the whole block), but in fact this flag is retained for backward compatibility.

The arrival of the first segment opens the reception session (“import session” in ION), while the arrival of the last, flagged as CP, triggers a report segment (RS) confirming that all data have been received. As the block is complete, its payload (one or more bundles) is passed to BP and the Rx buffer deallocated. RS reception confirms the CP and is in turn confirmed by a Report-ACK (RA). As the full block is confirmed, the transmission session is closed and the BP agent notified of successful delivery of the bundle(s) contained in the block.

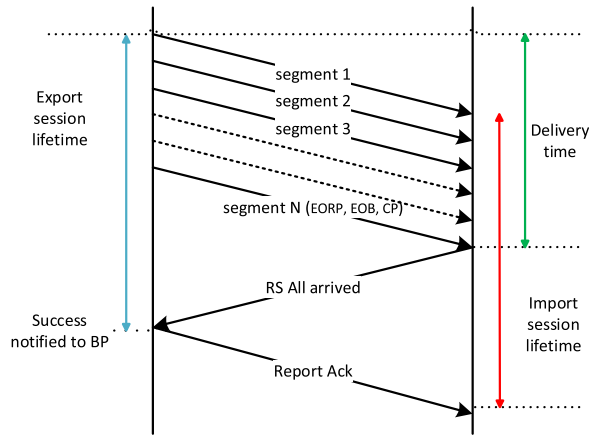


Fig. 1. Example of a red LTP session in ideal conditions (absence of losses). The radiation time has been expanded for clarity of representation.

All signaling segments (including data segments flagged as CP) except RA are protected against losses by a retransmission time-out (RTO) timer, equal to one round trip time (RTT) plus margin. It did not fire here because there were no losses.

In space links, block transmission time (also called “radiation time”) and block processing time are largely dominated by propagation delay. As a result, block delivery time is roughly equal to the propagation delay, ($1/2$ RTT), and both the transmission and reception session lifetimes are each equal to one RTT. This leads to the important conclusion that LTP is ideal in the absence of losses, because both delivery time and confirmed delivery time (i.e., for receiving transmitted data acknowledgment), are at their theoretical minima.

2) *Losses on Data Segments:* We are now ready to consider losses. For brevity, we will limit the analysis to losses on “pure” data segments, i.e., data segments not flagged as CP. If we examine Fig. 2, everything goes on as before but segments 2 and 3 are now lost. As a result, CP reception triggers only a partial RS, confirming all data but segments 2 and 3 (more precisely inside this RS we will have one “claim” for bytes of segment 1, and another for bytes of segments 4 to N). The two missing segments are then retransmitted, 3 being flagged as CP, and their arrival completes the block; from now on everything continues as in the ideal case of no losses.

The delivery time and duration of transmission and reception sessions are all increased by one RTT, the time necessary to perform one retransmission cycle. In the unlucky case of a loss on pure data retransmitted segments, the delay would now be twice RTT, and so on, with every additional retransmission cycle adding one RTT. Should the retransmission cycles exceed a certain number, the session would be cancelled by the sender and the BP notified of failure [9].

A key difference from TCP is that all unacknowledged data segments are retransmitted together, which limits the increase in delivery time to one RTT, independently of the number of segments lost (excluding, for simplicity,

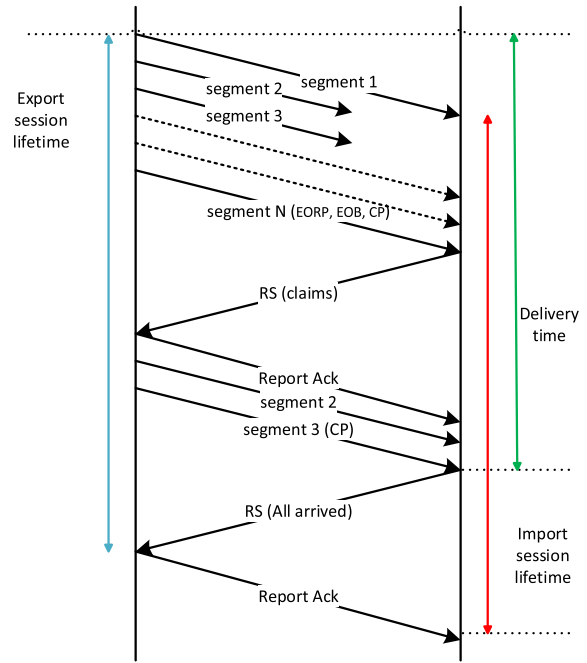


Fig. 2. Example of Red LTP session in the presence of losses on pure data segments (2 and 3). The delivery time increase is one RTT, independently of the number of losses.

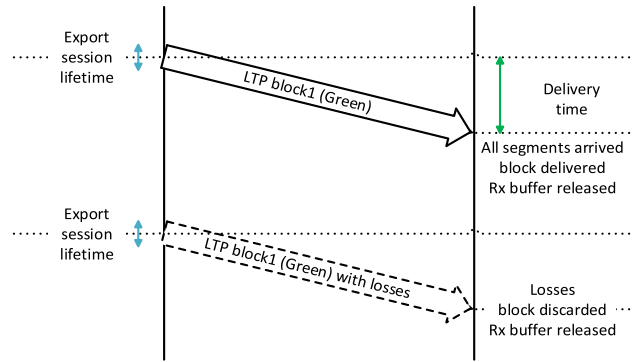


Fig. 3. Green sessions: (a) without losses (sess.1); (b) with losses (sess.2).

the case of consecutive losses). This is actually the theoretical minimum for automatic repeat reQuest (ARQ) based recovery, i.e., the best possible result for LTP. In the case of “pure” signalling segments, such as RSs or RAs, or mixed segments, i.e., CPs, the impact of losses would be worse. This has led to the development of an enhanced version of LTP, where signalling segments can be protected by a variable amount of replication [11]; this extension to standard LTP has already been introduced in ION and the same mechanism is incorporated in multicolor LTP.

B. Green Sessions (Unreliable Service)

Green sessions are much simpler, as they do not imply any signalling in the reverse direction.

1) *Standard Green Sessions:* When the transmission session starts (see Fig. 3), all data segments are sent, the last being flagged as EOB [9]. As retransmissions are now

barred, the Tx buffer is immediately released, the transmission session closed, and the BP immediately notified that the block has been “sent” (which means that it has been entirely passed to the protocol below, not necessarily already transmitted, two distinct concepts in the DTN world). By contrast to red, in green there is not feedback, as the service is unreliable, thus, as soon as the EOB is sent, the LTP sender notifies BP that it has terminated the transmission session. Unless other retention constraints apply (e.g., optional custodial duty in BP v6 [4], not considered in this article), the bundle can immediately be canceled by the BP sender.

At receiver side the arrival of the first segment opens the reception session, which is then closed on the arrival of the EOB, i.e., almost immediately. If all segments have arrived, the block is completed and the bundle can be delivered to the BP, otherwise the block is dropped. In both cases, the Rx buffer is immediately deallocated. Transmission and reception sessions are limited to the time strictly necessary to send or receive the block (“radiation time”, not represented in the figure as very short).

The behavior described differs from standard LTP procedures [9], [10] at Rx side, as these would prescribe individual green segments instead of a whole block being notified and passed to the upper protocol. We prefer the proposed multicolor LTP behavior for both theoretical (fragmentary units should not be passed to upper protocols as a general rule) and practical reasons (not to slow down receivers).

2) *Green “Corner Cases”*: Although green sessions are simple, attention must be paid to a couple of peculiar cases.

a) *Loss of the EOB*: The possible loss of green EOB, not covered in standard documents [9], [10], is tackled in multicolor LTP by the introduction of an intersegment timer, which allows for fast closure of the reception session and consequent deallocation of the Rx buffer, unless new segments of the same session arrive.

b) *Loss of the Full Block*: Another peculiar case is the possible loss of the full block. For green sessions this is a very brief story, as nothing would happen at either Tx or Rx side; by contrast, for red sessions the final CP would be retransmitted after one RTO and eventually either the full block would arrive or the session would be cancelled because of excessive retransmissions [9]. The only inconvenience with green sessions is that the upper protocol agent, after the block is notified as “sent”, would never know whether it has actually been received. Although the sole downside, this is of great impact and often undesirable for many applications.

C. Orange Sessions (“Notified” Service)

The introduction of orange sessions in multicolor LTP stems from the need to let the LTP (and in turn the BP) sender know the session result, as in red, i.e., whether the block has really been received, but without implying the burden of

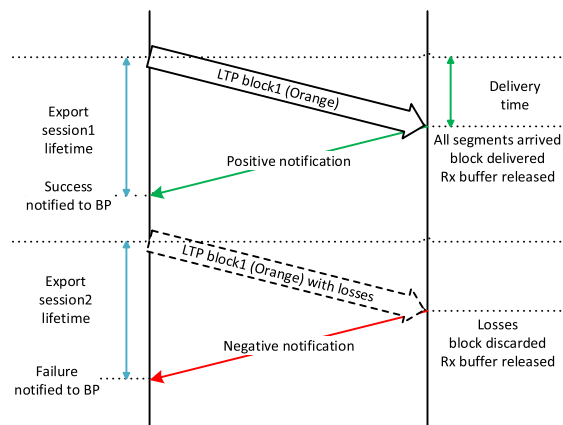


Fig. 4. Orange sessions: (a) without losses (sess.1); (b) with losses (sess.2).

segment retransmissions. Orange sessions resemble green plus result notification, although they are not exactly the same, as will shortly be evident. The rationale of notification is that in a few application scenarios the BP at transmitter side would like to know whether bundles passed to LTP have arrived or not, in order to take the most appropriate action, e.g., to drop or retransmit the bundle in the lost block. Although the choice is left by RFCs [4] and [6] to the BP implementation, let us anticipate here that ION always retransmits bundles, as will be shown in Section VII (Numerical results).

Orange sessions, like green sessions, minimize buffer allocation at both sender and receiver sides, as buffers can be deallocated as soon as the last segment is transmitted or received. This could be a major advantage on large BDP links (e.g., optical links to mars, see Section V-B).

1) *Standard Orange Sessions*: When the transmission session starts (see Fig. 4), all the segments of the orange LTP block are sent as orange data segments; the last flagged as orange EOB. As both “orange data not EOB” and “orange data EOB” segments are new, we propose to exploit two segment codes left unused in LTP RFC [9], namely 5 and 6. Because orange data segments are never retransmitted, the Tx-buffer is immediately deallocated, as in green, but by contrast to green the reception session must stay open for at least one RTT until arrival of the success/failure notification. An orange notification timer is set to cope with possible notification loss.

The arrival of the first data segment opens the reception session, which is then almost immediately closed by the EOB segment. If all segments have arrived [see Fig. 4(a)], the block is complete and the bundle can be delivered to BP; by contrast to green, a positive notification is now sent, signaling that full block reception; on its arrival the LTP sender notifies the BP of successful transmission. Otherwise [see Fig. 4(b)], the incomplete block is discarded and a negative notification sent to inform the LTP sender, which in turn will inform BP of failed transmission. The transmission session normally closes on notification arrival, i.e., after one RTT.

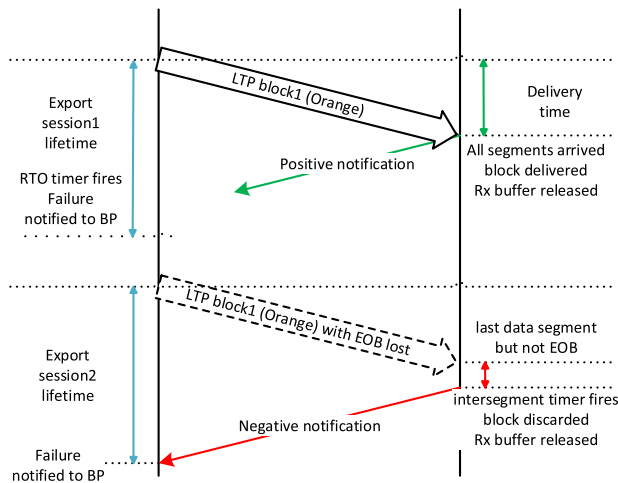


Fig. 5. Orange corner cases: (a) notification loss (sess.1); (b) EOB loss (sess.2).

2) *Orange “Corner Cases”*: a) *Loss of Notification (Positive or Negative)* If the notification is lost, the transmission session stays open until the orange notification timer fires, in which case the failure is notified to BP [see Fig. 5(a)]. The orange notification timer should fire after one RTO ($= RTT + \text{margin}$) from EOB dispatch. This is the same RTO timer already used for red signaling segments, thus, it introduces no additional complexity. This timer also prevents the transmission session lasting forever, should the full block be lost (in both cases no notification would arrive).

b) *Loss of Orange EOB* As with green, attention must be paid to the possible loss of the EOB. The main difference is that after the intersegment timer expires, here a negative notification is issued [see Fig. 5(b)].

In conclusion, let us stress once again that although similar to green, orange, like red, always notifies the upper protocol at sender side, e.g., BP, whether the PDU has been successfully delivered at next node, a crucial difference.

IV. SUMMARY OF NEW MULTICOLOR LTP FEATURES

The enhancements proposed by multicolor LTP are numerous but also significant. They are summarized in Table I and briefly discussed below. The most important is the fact that only monochrome blocks are allowed, which greatly simplifies protocol implementation and creates a one-to-one correspondence between QoS and sessions, a key feature in many respects. It also facilitates the introduction of new colors offering new services, such as orange, which is the second major difference with respect to standard LTP. These are the only two features that present compatibility issues with standard LTP: Orange sessions are obviously not supported by standard LTP and mixed sessions are no longer supported by multicolor LTP; all other combinations are interoperable.

Another significant difference is the introduction of timers, either to terminate green and orange sessions, should the EOB be lost, or as a safety measure to prevent a red session from stalling in unfortunate circumstances (interactions with flawed implementations, hijacked nodes, etc.).

TABLE I
Summary of Main Innovations

	Standard LTP	Multicolor LTP
Block colors	Monochrome or mixed (Red & Green parts)	Monochrome
Session color	Sessions have no color	The same color as the session block
Session QoS	Sessions have no QoS	The QoS of the color
Colors	Red, Green	Red, Green, Orange
Inter-segment timer (at Rx)	Not present	Present (Green and Orange)
Notification timer (at Tx)	Not present	Present (Orange)
Tx & Rx session timers	Not present	Present (Red)
Partial delivery (incomplete blocks)	Segment-by-segment delivery in Green	Whole blocks are passed (Red behavior is applied to all colors).
Link color	Links have no color	Links have a default color
Discretionary CPs and RS	Optional	Not used
RS acked	By RA	By RA and CP
RA of final RS	As other RAs	It always closes the session

A further difference concerns the interface between LTP and the upper protocol at reception side. Segment-by-segment notification and delivery is prescribed for green data in LTP standard specifications, while multicolor LTP passes only whole blocks, independently of color (in practice, red behavior applies to all colors, which also simplifies implementation).

Another difference is that the link to a proximate node must have a default color, i.e., a default QoS, to be used for PDUs (bundles) that do not require any special treatment. This default, if red or orange, can be overridden block-by-block, e.g., in the presence of special QoS flags in bundles to be sent; however, if the default is Green it cannot be overruled, in order to facilitate deployment on unidirectional links as shown later (see Section V-C).

Then, we have two minor improvements, already present in ION: first, the insertion of discretionary CPs and RSs [9] is no longer allowed (they would add complexity for limited benefits); second, RSs are also confirmed by CPs triggered by an RS (because they bear the RS number), and not only by RAs, not stated in [9].

A further improvement regards the reception of the RA triggered by the “final” RS (confirming the reception of all data in red sessions): this RA should close the ongoing session immediately as all data have been confirmed, even if other RSs are still pending, which may happen in peculiar circumstances.

V. APPLICATION SCENARIOS

A. Streaming Services

In this first example, we consider a video source, e.g., a camera on a space asset, sending a bundle stream to a MCC. These bundles are flagged with a flow label [23], to be recognized by intermediate nodes as belonging to the same flow. Let us assume that each bundle contains a number of consecutive video frames and is encapsulated in an LTP block on all hops between source and destination. There are two requirements: first, video frames should be delivered as soon as possible (in order, but with possible gaps), to be played in pseudoreal-time (pseudo because signal propagation delay is unavoidable and can be in the order of tens of minutes). Second, lost frames should be delivered later anyway, to provide the MCC with an integral version of the video stream. The bundle streaming service (BSS) and the BSS Protocol (BSSP) were proposed to meet these requirements [24], [25].

In brief, the BSS library agent at destination must distinguish between bundles arriving in order, whose content is to be viewed immediately, and delayed ones (because of retransmissions), whose content is to be passed to the database containing the video frames already received. By contrast, the BSSP is a convergence layer protocol, complementary to but also independent of BSS. Like the BSS agent, BSSP tries to distinguish between on time and delayed bundles. The former are sent as unreliable, i.e., green, for the sake of speed, the latter as reliable, i.e., red, to ensure delivery. The rationale is that on time bundles should continue to be forwarded as fast as possible. The critical point is that the BSSP transmitter must cope with the possible loss or incomplete reception of bundles sent as LTP Green blocks. To ensure retransmissions of lost data, positive notifications are sent by the BSSP receiver; if nothing arrives, a timer expires after an RTO and the data are resent, this time as red.

BSSP is ingenious but also cumbersome. With orange in mind, it is evident that the same service could be better provided by using the new color, which, in essence, offers the notified service that the BSSP notification tries to enforce. To cut a long story short, instead of an additional convergence layer, it would be enough to send the video flow bundles as orange LTP blocks the first time, and then, in case of notified failure, as red.

Apart from its intrinsic simplicity, using a new color instead of a different convergence layer offers many additional advantages: first, orange can also be used for other purposes (see below); second, bundles to be retransmitted with orange would necessarily be reprocessed by contact graph routing (CGR)/schedule-aware bundle routing [26], [27]. This dynamic routing algorithm (routing decisions are time dependent) could send the retransmitted bundles to a different neighbor if necessary (see Section VI-C), or, after one failure, drop those that can no longer be delivered in time. Moreover, bundles could be resent not only with a different color, but also with a different equivalent priority, i.e., they could be treated as bulk bundles to facilitate faster delivery of fresh bundles.

B. High Speed Space Links

To deal with losses there are two approaches, reactive, and proactive. The former uses ARQ to have retransmission from the transmitter node (possibly an intermediate node in DTN architecture), as in LTP red. The latter uses PL-FECs, which introduce redundancy packets, to recover from packet losses without resorting to retransmissions; note that these codes are not an alternative to usual FEC codes operating at physical layer, i.e., on bits, but are intended to complement them [12]–[14]. The best solution depends on the RTT, the bandwidth available and the product of the other two, i.e., the BDP. An analogy can help. A taxi driver in Los Angeles does not need to carry spare parts, because if the car breaks down it can rapidly be serviced, while a sensible guide taking tourists into the Sahara should; not because he could not use a satellite phone to call for help, but because it would take ages to receive any spare parts. On networks it is the same; this is why ARQ is advantageous on Internet, and much less appealing on interplanetary links, characterized by literally astronomical delays. Conversely, it is also true that spare parts require room. This is why on links with very limited bandwidth, ARQ can still be a reasonable choice, even when the RTT is relatively long (e.g., 2.5 s on lunar links). However, when the delay is huge and bandwidth abundant, as on future deep space optical links, the balance tends to be definitively in favor of PL-FEC solutions.

If we assume LTP on top of PL-FEC, several solutions are possible [19]: one could rely entirely on the loss recovery power of FEC, by using green, or adopt a mixed solution with red (ARQ plus PL-FEC). The former is the simpler, but in rare cases the green block could be lost due to a FEC decoding failure (essentially, when the lost LTP segments in a code word are more than the redundancy packets), thus, the service provided by green plus PL-FEC would still be theoretically unreliable, because of the lack of acknowledgments. If this is not acceptable, the second solution would be preferable: it retains the red ARQ mechanism, but LTP segment retransmissions would be used only as backup, to cope with (hopefully rare) FEC decoding failures; this time the service would be reliable.

The new orange color offers a third option. In the case of a FEC failure, it would simply notify the block loss to BP, leaving possible retransmissions to BP, like in the HSLTP proposal [20]. Once again there is no one-size-fits-all solution, but on interplanetary high-speed links the combined use of orange and PL-FEC seems preferable to the second solution, based on red. To understand why, we must consider that, at least in theory, ARQ (i.e., red) requires to buffer an amount of data equivalent to the BDP at both transmission and receiver sides. At transmitter, in order to resend lost LTP segments; at receiver, to reassemble the LTP block before bundle delivery to BP. The problem is that in interplanetary high speed links the BDP is really huge. For example, if we consider a one gigabit/s optical link between earth and mars, we would have a BDP in the range of 45:345 GB, i.e., several orders of magnitude larger

than on Internet. With orange this problem would not arise, as Tx and Rx buffers need only be large enough to host the current block.

In more detail, for fairness we need to remember that clever code design could alleviate the memory burden of Red, as ION, where LTP transmission buffers are basically avoided by using a “zero-copy” mechanism: the data segments to be retransmitted are rebuilt on the spot by LTP, using data stored by BP. This solution is clever, but neither general nor fast, thus, not well suited to high speed links. As for Rx buffers, in recent versions of ION they are deallocated not when a red reception session closes, but as soon as a red block is completed, i.e., immediately in case of no losses. Thus, in this ideal case, one Rx buffer (more precisely, one for each transmitting neighbor) would be enough, as with orange. However, in case of losses in the current block, the buffer would remain allocated to the on-going session for at least one RTT (several minutes on interplanetary links!), and reception could only continue by using a second buffer, and so on. The problem is that it is hard to estimate how many buffers should be allowed, and how large they should be, as this would depend on so many factors (channel coherence time, Tx speed, and bundle dimension). Once again, ION code alleviates the buffering problem, by allowing Rx buffers to be kept only partially in RAM. However, this slows down reception of incoming LTP segments and, thus, is not well suited to high speed links.

All these problems can be solved with orange on top of PL-FEC, as mentioned. In fact, thanks to orange only one Tx and Rx buffer would be necessary for each neighbor (as for green). Failures would be notified to BP, which could retransmit the bundle from scratch in a new session. This would not require any additional storage, as bundles must be retained in any case (on the BP database), until successful reception is notified to BP. Finally, all benefits of CGR reprocessing discussed in the streaming case (see Section V-A) would apply also in this case.

Of course, there would be a tradeoff with the bandwidth wasted, because instead of resending only the missing LTP segments, all bundles of the failed block would be retransmitted. The frequency of PL-FEC failures, however, can be kept under control by varying the amount of redundancy introduced [20], thus, the impact on bandwidth should be negligible on high BDP links.

C. Unidirectional Links

In ION the choice of LTP color is made on the basis of the ECOS “unreliable flag” set by the source. If this flag is set, green is chosen on all hops, otherwise red, also on all hops. This behavior is logical, but does not match well with the possible presence of unidirectional links on the path to destination. In fact, if a link is unidirectional there is no alternative to green, as direct confirmations are impossible, and in ION a bundle is allowed to pass on a unidirectional link only if flagged as unreliable, otherwise it is dropped. As a consequence, to cope with the presence of even a

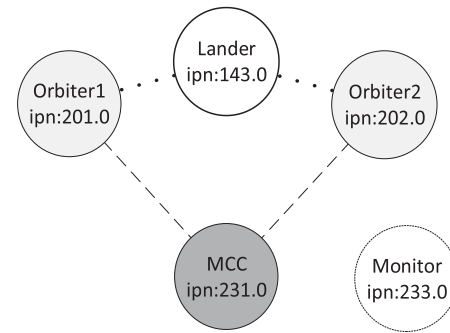


Fig. 6. DTN topology used in tests. DTN nodes are connected to each other via scheduled intermittent links (broken lines). The monitor node is connected to other nodes via a separated control network (not shown), to avoid any interference between data and testbed control traffic.

sole unidirectional link, the BP source is obliged to set the unreliable flag, which, however, implies that the service will be unreliable on all hops. The link default colors introduced by multicolor LTP solve this problem. If the source would prefer to have bundles transferred reliably wherever possible, i.e., on all hops but those with a unidirectional link, it should not set any flags, which would result in the use of default colors on all links, e.g., possibly red on all but the unidirectional one, thus solving the problem.

More generally, the choice of color should consider: 1) the QoS indicated by the bundle source, 2) any preferences of the BP agent (which for example could select a different color for retransmissions, as in Section V-A, and 3) the constraints of the physical layer, as here.

VI. SYSTEM CONFIGURATION

To illustrate multicolor LTP, we carried out a few tests, described in the following section. They aim to highlight novel characteristics, rather than provide any general performance evaluation, which would be beyond the scope of this article. To this end, we deliberately chose a minimal layout, an essential contact plan, and a basic experiment, repeated for the three different colors. In other words, the simplest possible experiment able to highlight the subtle differences between different colors when jointly analyzed with BP retransmissions and CGR.

A. Testbed Layout and Related Software

The DTN layout (see Fig. 6) consists of only four nodes, which although not pretending to be representative of any specific operational scenario, might be interpreted as a MCC, two Martian Orbiters, and a Martian Lander. A fifth node, the Monitor is used in the testbed only to collect Bundle Status Report (BSR), issued by other nodes when a bundle is received or delivered [4]. The monitor is connected to other nodes by means of a control network (not represented in the figure), which is used to manage all the experiments from the Monitor node and is isolated from the links used in experiments (broken lines in Fig. 6).

The protocol stack used in DTN nodes is summarized in Fig. 7. The DTNperf_3 application [28] runs as a client

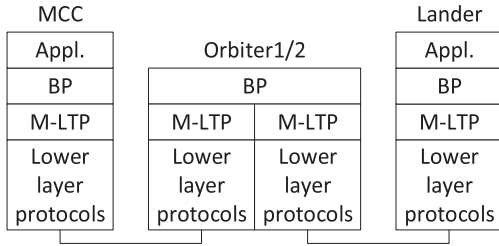


Fig. 7. Protocol stack of the DTN nodes involved in the experiments; the stack of the monitor is not reported but is the same as that of MCC and Lander; the BP application DTNperf runs on these nodes, respectively, as monitor, client, and server; M-LTP stands for multicolor LTP.

TABLE II
Reference Contact plan. #

#	Send. (ipn)	Rec. (ipn)	Start T. [diff](s)	End T. [diff](s)	Data Rate (byte/s)
1	231	201	+40	+90	1250000
2	231	202	+40	+90	1250000
3	201	143	+40	+50	1250000
4	201	143	+70	+80	1250000
5	202	143	+50	+60	1250000
6	202	143	+80	+90	1250000

on MCC and as a server on Lander; then we have the BP on top of multicolor LTP (M-LTP in the figure); below LTP there are the lower layer protocols (UDP, IP, and Ethernet in our testbed, while in a real deployment we would have specific CCSDS protocols. In our tests, we used BP v6, but the results would be the same as with v7, since we did not enable the custody option (no longer present in v7, see the just released RFC 9171 [6]).

Orbiter1 and Orbiter 2 have the same protocol stack as end-nodes, but they lack the application, as they are used only to route traffic, according to the BP store-and-forward procedures. This means that Orbiters, like all DTN nodes, must be able to store bundles in a local BP database for relatively long periods, to cope with intermittent connectivity. From the figure it is evident that BP is not an end-to-end protocol, as it is present on all DTN nodes, and that retransmissions between one node and the next can be carried out both at convergence layer (M-LTP, here) and at BP layer (e.g., when a session failure is notified to BP, as discussed).

All nodes are implemented by GNU/Linux Virtual Machines (VMs) running ION 3.7.4 and Unibo-LTP. VMs are managed by Virtualbricks [29], which also provides virtual switches to interconnect VMs, and channel emulators to introduce the desired amount of delay and loss on space links.

B. Contact Plan

Links are all scheduled intermittent and corresponding contacts are specified in Table II following the ION format. Contacts are unidirectional (they are defined as transmission opportunities from node A to node B), but as contacts

in opposite directions are assumed to be the same, for brevity only contacts in the uplink direction, from MCC to the Martian Lander, are listed. Contacts and “ranges” (propagation delays between nodes, not given in the table) are scaled down from minutes to seconds for convenience. Start and stop times are expressed differentially with respect to a reference time (e.g., ION startup).

The two contacts (1 and 2) between MCC and the two Martian Orbiters are relatively long (50 s) and are perfectly symmetrical to facilitate comprehension of routing decisions; the corresponding range is 5 s. Between Orbiter1 and Lander there are two short contacts (3 and 4, 10 s), whose range is limited to 1 s, as orbiters are supposed to be close to the Lander. Contacts from Orbiter2 (5 and 6) are the same, but delayed by 10 s.

C. Tests

Our experiment consists of sending 20 bundles of 50 kB each from MCC to the Lander; each bundle is encapsulated in an LTP block, and each block divided into 49 data segments of 1024 B. On the first hop, on MCC to Orbiter links we have a moderate Packet Loss Rate (PLR = 0.5%) and a 5 s one-way delay equal to the mentioned “range”, giving a 10 s RTT. On the second hop, on orbiters to Lander links (presumably less challenged because of the shorter distance), the losses are assumed negligible (PLR = 0) and the delay is only 1 s (RTT = 2 s). The experiment is repeated three times, by setting a different color on the first hop, while on the second hop the default color is constantly set to green.

VII. NUMERICAL RESULTS

The following analysis is largely based on BSRs collected by DTNperf monitor, complemented when necessary by logs provided by Unibo-LTP (not given here for brevity). We will examine the three colors sequentially to highlight the differences.

A. Red on Earth to Mars Links

The 20 bundles destined for Lander are generated at a regular pace on MCC (“Generated” time series in Fig. 8). Once generated, each bundle is processed by CGR to find the best route to destination. Without entering into CGR details [27], we would remind the reader that a CGR route does not consist of an ordered series of intermediate nodes, as in Internet, but of contacts. When the route for the first bundle is computed, the best solution (i.e., the combination of contacts that ensure fastest delivery) is provided by contacts (1, 3). As a result, the first bundle is put in a queue for node 201 (the destination node of contact 1, i.e., Orbiter 1), to be stored until contact 1 starts, at 40 s. In the meantime, the other 19 bundles are put in the same queue, because the best route for them all is still the same.

When contact 1 starts, all these 20 bundles are passed to LTP and sent to 201 in 20 consecutive, independent red sessions. If all segments of a session arrive, the bundle in the LTP block is passed to BP agent on node 201 (all bundles but 3, 8, 9, 11, and 18; see “Rcv at 201” time series in Fig. 8,

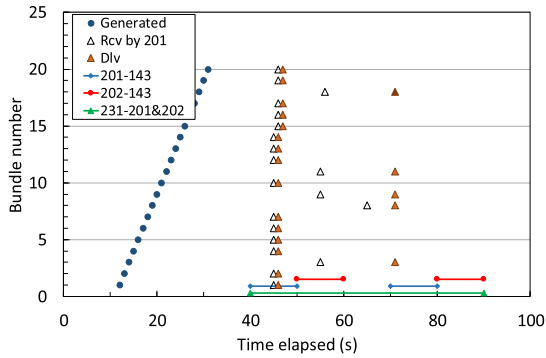


Fig. 8. Multicolor LTP red on earth to Martian Orbiters hops (231–201 and 231–202; green on orbiters to Lander hops (201–143 and 202–143). Twenty bundles, 50 kB each, from MCC to Lander. Bundles 3, 8, 9, 11, and 18 are delivered 25 s later than expected, on the second 201–143 contact instead of the first. 15 Tx buffers allocated for 1 RTT, 4 for 2 RTTs, 1 for 3 RTTs. 4 Rx buffers allocated for 1 RTT, 1 for 2 RTTs.

markers at 45 s). Bundles that arrived successfully are: a) processed by CGR, which confirms contact 3 (the first from 201 to 143) as the best choice; b) then queued for node 143 (the Lander) and immediately sent, as contact 3 is already active; and c) finally delivered 1 s later (“Dlv” time series at about 46 s).

After the first round, the five uncompleted blocks have to be kept in Rx buffers on 201 waiting for segment retransmissions. One retransmission cycle is generally enough to complete the block, but for the most unfortunate two cycles are necessary, because of consecutive losses. As a result, 4 bundles (3, 9, 11, and 18) are passed to the BP agent on 201 with one RTT of delay (“Rcv at 201”, markers at 55 s), and one (bundle 8) with two RTTs (“Rcv at 201”, the sole marker at 65 s). For them all, however, the problem is that retransmission delay means they have “missed the connection” to contact 3, closed at 50 s. Therefore, when CGR processes the delayed bundles on 201, the best route now consists of contact 4 (the second 201–143 contact) and finally all these delayed bundles are delivered at 71 s, after contact 4 starts.

To summarize, we can observe that with red on first hop, all bundles arrive at destination, 15 in the expected time, 5 delayed by 25 s because of a missed contact, which clearly proves the impact of segment retransmission on routing, an often neglected issue. To allow for segment loss recovery, Rx buffers on node 201 were occupied for one RTT by bundles 3, 9, 11, 18, and for two RTTs by bundle 8. Tx buffers were in use for one RTT for the 15 “lucky” bundles; for two RTTs by bundles 3, 9, 11, 18; for three RTTs by the unluckiest, bundle 8. This price may become significant or even unsustainable on large BDP links, as previously discussed.

Using PL-FEC under LTP Red can greatly reduce the chances of incomplete block reception, i.e., of retransmission-induced delays for Reds. However, it can neither alleviate nor solve the problem of buffering at Tx side, as it would not eliminate the possibility of retransmissions in the case of FEC decoding failures. An additional drawback is that the interface between PL-FEC and Red is

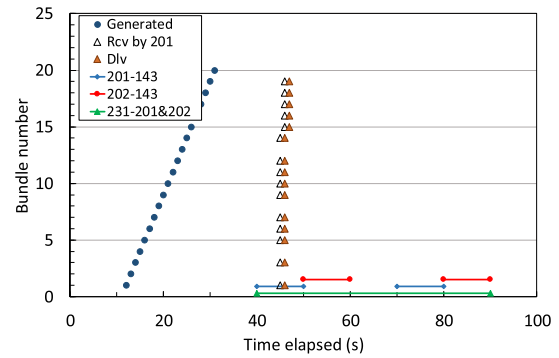


Fig. 9. As in Fig. 8, but green on earth to mars hops. All bundles delivered at 46/47 s but 2, 4, 8, 13, and 20, which are lost on the first hop. Only one Tx and one Rx buffer occupied for marginal time on the first hop.

quite complex, essentially because red signaling segments are difficult to manage and, among other things, require an aggregation timer, which is really problematic to set [19], [20].

B. Green on Earth to Mars Links

With green on the first hop we can observe (see Fig. 9) that all but five bundles are delivered in one burst as with Red (“Dlv” time series at 46/47 s). Unlike with red, the others (2, 4, 8, 13, and 20) never arrive, due to LTP segment losses on the first hop (uncompleted green blocks are discarded). This obvious disadvantage is partially compensated for by the fact that only one Tx and one Rx buffer were necessary, and that they were used for only “radiation time”, a negligible fraction of the RTT.

This test makes it clear that green should not be used when there is a significant probability of at least one segment loss in a block. Unfortunately, this probability depends not only on the channel PLR, but also on block length, i.e., on bundle dimension. This harmful coupling of LTP performance and bundle dimension can be avoided, as in red, by making use of PL-FEC under LTP. Although this scheme (Green+PL-FEC), cannot provide theoretical “reliability,” because it lacks feedback, it is the best we can do on unidirectional links.

C. Orange on Earth to Mars Links

Moving on to orange (see Fig. 10), we can see that once again 15 bundles are delivered when expected, at 46 s, while 5 (2, 7, 8, 17, and 19) are delayed. From CGR and Unibo-LTP logs we can see that the original blocks containing the five delayed bundles, having suffered at least one loss, were immediately dropped, as in green. However, this time orange negative notifications are sent to the LTP transmitter on node 231, and then BP is notified of failure (at 50 s). This always triggers a retransmission at BP layer in ION, thus, these bundles are reprocessed by CGR, which discovers that the original route (1, 3) is no longer viable (the expected arrival time at 201 would be at 55 s, i.e., 5 s after contact 3 stop time). As the new best route consists of contacts (2, 5), these bundles are sent to node 202 instead of 201. Four bundles

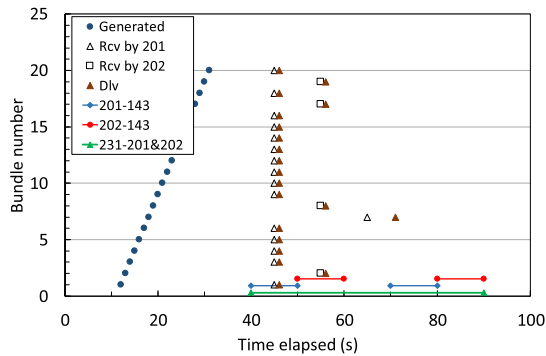


Fig. 10. As in Figs. 8 and 9, but orange on earth to mars hops. All bundles delivered, but 2, 8, 17, and 19 delayed by about 10 s, bundle 7 by 25 s. One Tx and one Rx buffer occupied for marginal time on first hop.

(2, 8, 17, and 19) arrive when contact 3 is active (“Rcv by 202” markers at 55 s), thus, they are immediately forwarded to Lander and delivered there at about 56 s, i.e., one RTT later than expected. Bundle 7 is the most unfortunate, as its encapsulating LTP block to 202, affected by at least one loss, had to be discarded once again. From the logs we can see that the bundle is retransmitted by BP a second time, at about 65 s, processed by CGR, which replaces the old (2, 5) with the new (1, 4) route, the best at this late time. The bundle is, thus, sent to 201, correctly received (“Rcv” by 201 marker at 60 s), stored, and finally delivered to the Lander when contacts 4 starts (“Div” marker at 71 s).

To summarize, with orange all bundles are delivered, as in red. The additional delay due to losses on the first hop is limited to one RTT for four out of five delayed bundles (instead of 2.5 RTTs as in red), thanks to CGR reprocessing retransmitted bundles. For one bundle the delay is 2.5 RTTs as in red, due to two consecutive failures. Finally, only one Tx and one Rx buffer were necessary on the first hop, and they were used for a negligible amount of time, as in green.

Orange should not be used when BP retransmissions are performed as a result of a failure notification, as in ION, unless the probability of consecutive block losses is low, which poses the same problem of LTP performance depending on bundle dimension, an issue already pointed out for green, but also present in red. Again, PL-FEC is the best solution to this problem. Moreover, thanks to the lack of signaling segments in the forward direction, the PL-FEC interface for orange is much simpler than that for red as aggregation timers for signaling segments are no longer required.

For video streaming, it would also be possible to resend bundles in a different color, i.e., in red. This mechanism could be coupled with PL-FEC to achieve both theoretical reliability and (usually) fast delivery for video streams.

VIII. CONCLUSION

In this article, multicolor LTP was presented in full detail. It consists of a dozen enhancements, the most important of which are monochrome sessions, orange, offering a “notified” service, and the individual link default colors. Possible application scenarios include BP video streams,

networks with unidirectional legs, and large-BDP optical links, such as those envisaged for future interplanetary connections. The analysis of a simple scenario offered insight into complex LTP mechanisms and interactions with routing and PL-FEC. Multicolor LTP is almost completely compatible with standard LTP; it has already been implemented as an ION plugin, released as free software. Multicolor LTP enhancements are to be proposed to CCSDS SIS-DTN working group for possible inclusion in new LTP specifications (LTPv2).

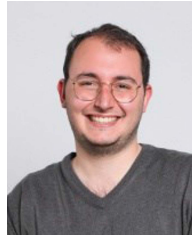
ACKNOWLEDGMENT

The authors would like to thank Scott Burleigh, ex NASA-JPL, for his more than a decade-long support with ION and for his valuable discussions and criticisms of new multicolor LTP features, including suggestions for improving orange notifications.

REFERENCES

- [1] S. Burleigh *et al.* Delay-tolerant networking: An approach to interplanetary internet *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003, doi: [10.1109/MCOM.2003.1204759](https://doi.org/10.1109/MCOM.2003.1204759).
- [2] V. Cerf *et al.* Delay-Tolerant networking architecture Internet RFC 4838, Apr. 2007. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4838.txt>
- [3] CCSDS 734.0-G-1 Green Book Rationale, scenarios, and requirements for DTN in space 2010. [Online]. Available: <https://public.ccsds.org/Pubs/734x0g1e1.pdf>
- [4] K. Scott and S. Burleigh Bundle protocol specification Internet RFC 5050, Nov. 2007. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5050.txt>
- [5] CCSDS 734.2-B-1 Blue book Bundle protocol specification 2015. [Online]. Available: <https://public.ccsds.org/Pubs/734x2b1.pdf>
- [6] S. Burleigh, K. Fall, and E. Birrane Bundle protocol version 7 Internet RFC 9171, Jan. 2022. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc9171.html>
- [7] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese Delay- and Disruption-Tolerant networking (DTN): An alternative solution for future satellite networking applications *Proc. IEEE*, vol. 99, no. 11, pp. 1980–1997, Nov. 2011, doi: [10.1109/JPROC.2011.2158378](https://doi.org/10.1109/JPROC.2011.2158378).
- [8] M. Ramadas, S. Burleigh, and S. Farrell Licklider transmission protocol – motivation Internet RFC 5325, Sep. 2008. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5325.txt>
- [9] M. Ramadas, S. Burleigh, and S. Farrell Licklider transmission protocol – specification Internet RFC 5326, Sep. 2008. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5326.txt>
- [10] CCSDS 734.1-B-1 Blue book Licklider transmission protocol (LTP) for CCSDS 2015. [Online]. Available: <https://public.ccsds.org/Pubs/734x1b1.pdf>
- [11] N. Alessi, S. Burleigh, C. Caini, and T. de Cola Design and performance evaluation of LTP enhancements for lossy space channels *Int. J. Satell. Commun. Netw.*, vol. 37, pp. 3–14, 2019.

- [12] T. De Cola and M. Marchese
Reliable data delivery over deep space networks: Benefits of long erasure codes over ARQ strategies
IEEE Wireless Commun., vol. 17, no. 2, pp. 57–65, Apr. 2010, doi: [10.1109/MWC.2010.5450661](https://doi.org/10.1109/MWC.2010.5450661).
- [13] T. de Cola, E. Paolini, G. Liva, and G. P. Calzolari
Reliability options for data communications in the future deep-space missions
Proc. IEEE, vol. 99, no. 11, pp. 2056–2074, Nov. 2011, doi: [10.1109/JPROC.2011.2159571](https://doi.org/10.1109/JPROC.2011.2159571).
- [14] CCSDS 131.5-O Orange Book Erasure correcting codes for near earth and deep space communications
2014. [Online]. Available: <https://public.ccsds.org/Pubs/131x5o1.pdf>
- [15] L. Shi *et al.*
Integration of Reed-Solomon codes to licklider transmission protocol (LTP) for space DTN
IEEE Aerosp. Electron. Syst. Mag., vol. 32, no. 4, pp. 48–55, Apr. 2017, doi: [10.1109/MAES.2017.160118](https://doi.org/10.1109/MAES.2017.160118).
- [16] J. Jiao *et al.*
Reliable deep-space file transfers: How data transfer can be ensured within a single round-trip interval
IEEE Veh. Technol. Mag., vol. 12, no. 4, pp. 86–94, Dec. 2017, doi: [10.1109/MVT.2017.2746478](https://doi.org/10.1109/MVT.2017.2746478).
- [17] G. Yang, R. Wang, S. C. Burleigh, and K. Zhao
Analysis of licklider transmission protocol for reliable file delivery in space vehicle communications with random link interruptions
IEEE Trans. Veh. Technol., vol. 68, no. 4, pp. 3919–3932, Apr. 2019, doi: [10.1109/TVT.2019.2897305](https://doi.org/10.1109/TVT.2019.2897305).
- [18] T. de Cola, H. Ernst, and M. Marchese
Performance analysis of CCSDS file delivery protocol and erasure coding techniques in deep space environments
Comput. Netw., vol. 51, no. 14, pp. 4032–4049, Oct. 2007.
- [19] N. Alessi, C. Caini, T. de Cola, and M. Raminella
Packet layer erasure coding in interplanetary links: The LTP erasure coding link service adapter
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 403–414, Feb. 2020, doi: [10.1109/TAES.2019.2916271](https://doi.org/10.1109/TAES.2019.2916271).
- [20] N. Alessi, C. Caini, A. Ciliberti, and T. de Cola
HSLTP—An LTP variant for high-speed links and memory constrained nodes
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 4, pp. 2922–2933, Aug. 2020, doi: [10.1109/TAES.2019.2958190](https://doi.org/10.1109/TAES.2019.2958190).
- [21] S. Burleigh
Interplanetary overlay network (ION) an implementation of the DTN bundle protocol
In *Proc. 4th IEEE Consum. Commun. Netw. Conf.*, 2007, pp. 222–226.
- [22] ION code and manual.
[Online]. Available: <http://sourceforge.net/projects/ion-dtn/>
- [23] S. Burleigh and F. Templin
Bundle protocol extended class of service (ECOS)
Internet draft, May 2021. [Online]. Available: <https://datatracker.ietf.org/doc/draft-burleigh-dtn-ecos/>
- [24] S.-A. Lenas, S. C. Burleigh, and V. Tsoussidis
Bundle streaming service: Design, implementation and performance evaluation
Trans. Emerg. Telecommun. Technol., vol. 26, no. 5, pp. 905–917, 2015.
- [25] CCSDS 730.2-G-1 Concepts and Rationale for Streaming Services over Bundle Protocol,
CCSDS Green Book, Sep. 2018. [Online]. Available: <https://public.ccsds.org/Pubs/730x2g1.pdf>
- [26] CCSDS 734.3-B-1, Schedule-Aware bundle routing
Blue Book, Jul. 2019. [Online]. Available: <https://public.ccsds.org/Pubs/734x3b1.pdf>
- [27] C. Caini, G. M. De Cola, and L. Persampieri
Schedule-Aware bundle routing: Analysis and enhancements
Int. J. Satell. Commun. Netw., vol. 39, pp. 237–249, Dec. 2020, doi: [org/10.1002/sat.1384](https://doi.org/10.1002/sat.1384).
- [28] C. Caini, A. d’Amico, and M. Rodolfi
DTNperf_3: A further enhanced tool for Delay-/Disruption-Tolerant Networking performance evaluation
In *Proc. IEEE Glob. Commun. Conf.*, 2013, pp. 3009–3015, doi: [10.1109/GLOCOM.2013.6831533](https://doi.org/10.1109/GLOCOM.2013.6831533).
- [29] P. Apollonio, C. Caini, M. Giusti, and D. Lacamera
Virtualbricks for DTN satellite communications research and education
In *Proc. PSATS*, Genoa, Italy, Jul. 2014, pp. 1–14, doi: [10.1007/978-3-319-47081-8-7](https://doi.org/10.1007/978-3-319-47081-8-7).



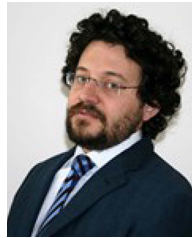
Andrea Bisacchi received the Degree in computer science engineering, with a thesis on the refinement of the ION implementation of ECLSA (an Error Correction based Link Service Adapter for LTP) and the master’s degree in computer science engineering “*summa cum laude*” (with a thesis on the design and implementation of *Unibo-LTP*) from the University of Bologna, Bologna, Italy, in October 2018 and March 2021, respectively.

His main research interests include Delay-/Disruption-Tolerant Networking and related protocols.



Carlo Caini (Member, IEEE) received the master’s degree in electrical engineering “*summa cum laude*” from the University of Bologna, Bologna, Italy, in 1986.

Since 1990, he has been with the Department of Electrical, Electronics and Information Engineering (ex Department of Electronics Computer Science and Systems), University of Bologna, where is working as Associate Professor of Telecommunications. He has coauthored about 100 papers on international conferences and journals; he has supervised the development of many free software components for these environments. His current research interests include Delay-/Disruption-Tolerant Networking and Transport Protocols for Space and Satellite Networks.



Tomaso de Cola (Member, IEEE) was born in Manosque, France, in April 28, 1977. He received the “Laurea” degree (with hon.) in telecommunication engineering, the Qualification degree in professional engineer, and the Ph.D. degree in electronic and computer engineering, robotics and telecommunications all from the University of Genoa, Genoa, Italy, in 2001, 2002, and 2010, respectively.

From 2002 until 2007, he was with the Italian Consortium of Telecommunications, 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 DVB standards, CCSDS protocols and testbed design. He has coauthored more than 100 papers, including international conferences and journals. His main research interests include TCP/IP protocols, satellite networks, transport protocols for wireless links, and interplanetary networks, as well as delay tolerant networks.

Dr. de Cola was a chair of the Satellite and Space Communications technical committee within IEEE Communication Society from 2017 to 2020; he has also been serving as chair of Space Internetworking Group of Consultative Committee for Space Data Systems since 2020.

Open Access provided by ‘Alma Mater Studiorum - Università di Bologna’ within the CRUI CARE Agreement