

Aggressive and Proactive LTP Control Signal Handling for Minimal Session Delivery Time: RTT Rules the World

Cheol Hea Koo and Scott C. Burleigh

Abstract—Several elements characterize deep space communications, including weak signal-to-noise-ratio (SNR), high bit error rate (BER), asymmetric channel bandwidth, and long propagation delay. In deep space missions, one-way light time (OWLT) is relatively much longer than in near-Earth missions. OWLT dominates data delivery completion time during a Licklider transmission protocol (LTP) transaction, making other communication elements relatively insignificant. As delay-/disruption-tolerant networking (DTN) technology plays a major role in communication for space exploration missions, especially Artemis missions in a cislunar environment, the performance of the LTP “convergence layer” protocol grows more important; reducing the time required to close an LTP transmission session will be increasingly critical. LTP session completion is crucial for mission operation because it must be bounded to support real-time operation. This study found that the LTP session closing time can be unacceptably long when link performance is in the BER range of 10^{-5} to 10^{-6} , which is commonly experienced in space exploration communications. This paper presents an aggressive and proactive LTP control signal handling mechanism, conforming to the published LTP standard, that can reduce the latency of LTP session closing time at the cost of somewhat diminished goodput ratio. By applying this scheme in tests configured for segment size 2000 and BER 10^{-6} , 99.67% of LTP sessions closed within 5 OWLTs, while similar tests in which this scheme was omitted which has 8.39% of shorter session closing time and only 4% chances of exceeding 5 OWLTs against a case non-applying it. Through numerical models and simulations, we show that the overhead is marginally acceptable and can contribute to better QoS over DTN operation in cislunar or deep space missions by bounding the LTP session closing time.

Index Terms—Bit-error rate, DTN, Licklider transmission protocol, one-way light time, round-trip time.

I. INTRODUCTION

DELAY-/disruption-tolerant networking (DTN) technology will be used in communication among the ground facility, relay communication assets, and landers/rovers/science instruments/astronauts on the Moon in Artemis missions [1].

Manuscript received November 14, 2022 revised February 19, 2023; approved for publication by Kyung-Joon Park, Division 3 Editor, May 3, 2023.

This study is supported by an internal research and development project in Korea Aerospace Research Institute, NTIS control number 1711196029.

C. H. Koo is with Korea Aerospace Research Institute (KARI), 169-84, Gwahak-ro, Yuseong-gu, Daejeon, 34133, Republic of Korea, e-mail: chkoo@kari.re.kr.

S. C. Burleigh was with Jet Propulsion Laboratory, Pasadena, CA 91109, USA and is with the InterPlanetary Networking Special Interest Group (IPN-SIG), e-mail: sburleig.sb@gmail.com.

C. H. Koo is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2023.000020

Communication systems incorporating DTN technology consist of multiple layered service protocols. Licklider transmission protocol (LTP) and bundle protocol (BP) are the core protocols required to use DTN technology in nominal space communication systems; BP is an overlay network protocol that relies on underlying “convergence layer (CL)” protocols such as LTP for data transmission, and LTP in turn relies on further underlying “link service” protocols for the exchange of subnetwork traffic. Unlike in the ground environment where transmission control protocol convergence layer (TCPCL), user datagram protocol convergence layer (UDPCL), Saratoga, and so on are available underneath BP, LTP may be the only option at the CL for DTN operation in space. Because the operation of LTP is based on an acknowledgment (ACK) and retransmission mechanism to ensure reliable data delivery, the total time required to “close” an LTP block transmission session is uncertain; it depends on the time required to complete each “round-trip” protocol exchange, which is a function of the signal propagation delay or “one-way light time” (OWLT) between LTP protocol endpoints, and on the number of such exchanges that are required to complete data delivery, which is a function of the bit-error rate (BER). When OWLT is large and BER is high, data delivery latency may become unacceptably high and the net data delivery rate of LTP may be sharply reduced.

Note that LTP retransmission is reactive: Data are retransmitted upon either (a) reception of a negative ACK (report) segment identifying lost segments or (b) lapse of the time interval within which reception of a positive or negative ACK is expected. Since neither of these events can occur before the original transmission time plus one RTT, the delay in LTP’s delivery of a given data block increases by multiples of RTT as the segment loss rate increases and the closing time in LTP’s delivery is not bounded. This effect may not be tolerable for applications requiring a minimal delay in data delivery.

To minimize variation in LTP session closing time, an aggressive and proactive LTP control mechanism has been devised to minimize the chance of retransmission; however, erasure coding during transmission can serve the same purpose. Therefore, we compared our proposed method and erasure coding.

Loss of an LTP signaling segment (such as a report segment (RS), functioning as an ACK of data reception) necessarily increases LTP session completion time by one RTT, so it is desirable to protect LTP signaling information from losses [2], [3]. Wu et al. [4] showed that a discretionary

Creative Commons Attribution-NonCommercial (CC BY-NC).

This is an Open Access article distributed under the terms of Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided that the original work is properly cited.

RS on an LTP transaction benefits file delivery completion time. Yu et al. [5] provided a detailed analytic computational and behavioral model of RTT in DTN operation. Lengthy delay over deep space communication dominates latency of packet/block delivery completion time regardless of block size [6]. However, large block size on a lossy channel negatively impacts block delivery completion time due to higher chances of retransmission [7]. Zhang et al. [8] showed that channel symmetry does not impact the protocol performance for long-link disruption in space communication. As frequent retransmissions of LTP segmentation blocks are requested, block delivery completion time increases and negatively impacts goodput performance [9].

Researchers studied erasure coding as a means of minimizing the chance of retransmission as this forward error correction (FEC) can recover a corrupted packet or segment in situ. Low density parity check (LDPC) erasure coding can at best provide about 1.4 dB coding gain over Reed-Solomon & convolution coding, resulting in about 10 times better BER in a range of approximately $10^{-5} - 10^{-7}$ [10]. Consultative committee for space data systems (CCSDS) recommends type-I hybrid automatic retry request (ARQ) and type-II hybrid ARQ [3]. When system BER is higher than 5×10^{-6} , and after adapting Reed-Solomon rate of 1.14 channel coding, the LDPC rate of 1.5 resulting in 5×10^{-7} BER has better performance in terms of goodput and fast LTP session closing time. When system BER is lower than 10^{-6} , there is no significant improvement using an LDPC rate of 1.5 since there is non-negligible goodput reduction and no QoS enhancement even if BER is improved to 10^{-7} .

Adapting erasure coding such as Reed-Solomon or LDPC on LTP operation is advantageous when transmitting larger blocks on channels with higher error rates [11]. Erasure coding complements the LTP's ACK mechanisms by providing corrupted packet recovery capability; however, the retransmission function still has to be provided in the LTP layer [12].

As mentioned above, bounding session closing latency during an LTP transaction is crucial for real-time operation in a deep space mission. This study focuses on an implementable technique for controlling the LTP session closing time in a bounded range. A numerical model is developed, and a simulation is performed to verify the proposed technique. The simulation results validate the effectiveness of the proposed technique for deterministically limiting LTP session closing time. Major keywords and important terminologies used in this study are listed in Table I.

This paper is structured as follows: Section II presents a summary of DTN in deep space missions. Section III discusses the role of LTP at the CL. Section IV introduces the core concept of the study and proposed technique. Section V presents the modeling of numerical methods and simulation results. Finally, Section VI gives the conclusion of the study.

II. DTN IN DEEP SPACE MISSIONS

Interplanetary overlay network (ION) is a DTN protocol software suite developed by Jet Propulsion Laboratory (JPL)

TABLE I
List of acronyms and abbreviations, and description.

Acronyms Abbreviations	Description
BP	Bundle protocol Upper-layer of DTN, providing network OSI layer
CCSDS	Consultative committee for space data systems International committee, handling space data communication standardization in space agencies, institutes and companies
CFDP	CCSDS file delivery protocol Protocol for file delivery service, can be operable with and without DTN
CL	Convergence layer Lower-layer of DTN, providing reliable data link OSI layer for BP
CLA	Convergence layer adapter Software implementation that supports the designated CL protocol, e.g., UDPCLA is typically composed of UDP/IP socket programming with remote node for receiving and sending interface service independently
CP	Checkpoint One of LTP signals generated by LTP sender, requesting a response by LTP receiver on current LTP session's transaction
DS	Data segment Segment delivering data from LTP sender to LTP receiver, can hold CP and RAS
DTN	Delay-/disruption-tolerant networking A networking theory and communication technology for space communication mesh network
EOB	End-of-block Last block of DS marked as final on current LTP session's transaction
LTP	Licklider transmission protocol Point-to-point reliable data delivery service, one of CL services for DTN
OWLT	One-way light time Required time of one-way transmission from node to node
RAS	Report-ACK segment ACK of the RS from LTP receiver
RS	Report segment Response action to the CP from LTP sender, to be generated by LTP receiver
RTT	Round-trip time Required time of communication turn-around between sender and receiver
TCPCCL	Transmission control protocol CL CL functionality supported by TCP/IP
UDPCL	User datagram protocol CL CL functionality supported by UDP/IP

and other cooperative partners to implement DTN technology and has been kept updated since 2007 [13]. In 2008, deep impact network experiment (DINET) project tested DTN technology in a space environment [14]. In 2009, a science instrument payload, commercial generic bio processing apparatus (CGBA)-5, which has DTN interface capability, was installed on the international space station (ISS). A series of DTN tests were performed between the ground and ISS. DTN technology was tested on the CGBA-5/ISS to check its availability for supporting network communication using DTN in space [15]. During this test, space experiment data were successfully downlinked to the ground via a DTN network consisting of ION (as an ISS's payload), DTN2 (as a gateway), and another ION (as a ground science user). However, there were several minutes of disruption during data handover via tracking and data relay satellite system (TDRSS) [16]. Test

scenarios on the ISS consisted of nominal DTN and CCSDS file delivery protocol (CFDP) file delivery tests [17].

In 2013, a DTN experiment with virtual relay on the lunar atmosphere dust and environment explorer (LADEE)/lunar laser communications demonstration (LLCD) payload was performed. It was the first experiment using BP and LTP over a spacecraft's optical link [18]. Despite temporary disruption by clouds during laser transmission, retransmission was successfully performed by DTN protocols without any operator intervention [19].

A study for lunar communications and navigation architectures has been performed to support the long-term lunar mission, resulting in the LunaNet architecture [20]. The LunaNet architecture consolidates space communication technology with delay and DTN technology to provide a more robust, flexible, and interoperable lunar communication architecture for long, intensive, and sustainable lunar activity and missions such as the Artemis mission. In 2018, the interagency operations advisory group (IOAG) announced a lunar planetary network architecture that proposes that most robotic assets in lunar orbit and lunar surface be DTN nodes, including lunar relay orbiters [21].

International communities and working groups have been organized to study and standardize DTN technology in space. CCSDS is an international community supporting and developing standardization of space communication technology, and a CCSDS DTN working group is chartered for handling DTN standardization issues. Interplanetary networking special interest group (IPNSIG), a chapter of the Internet Society, is a technical expert group organized internationally with space agencies, academies, and companies to publicly discuss DTN's technical issues and promote its use [22].

III. LTP AS A CL

LTP functions within the DTN protocol stack as an ARQ mechanism, similar to TCP. It differs from TCP in several ways; in particular, LTP effects retransmission between forwarding points within the network rather than between the communication source and destination endpoints. Because the RTTs characterizing pairs of topologically adjacent forwarding points may vary dramatically, end-to-end data delivery time may often be sharply reduced by retransmitting lost data only between forwarding points that are separated by very short distances (e.g., between an orbiter and lander at Mars) rather than between the ultimate source and final destination.

The increases in end-to-end data delivery latency resulting from reactive retransmission over lengthy RTTs are exacerbated by degraded link performance. As the BER on a communication link increases, the likelihood of data loss – reinitiating the transmission cycle – increases, and end-to-end latency is multiplied. Minimizing BER is therefore a necessary, though not sufficient, a contributor to satisfactory service for applications requiring a minimal delay in data delivery.

One key strategy for minimizing BER is to operate spacecraft in an environment characterized by a very high signal-to-noise ratio [23]. This is made possible by applying relevant

design parameters, such as minimal radio interference, ample transmission power, accurate antenna alignment, and precise attitude control [24]. Signal coding, attaching parity bits to transmitted frames to enable automatic bit-error detection and correction, is likewise vital.

The DTN BP relies on the data transmission services provided by the aggregated protocols below BP in the stack, called the “CL” and BP implementation modules invoking the services of these protocols to send and receive bundles are termed “*convergence layer adapters*” (CLA). LTPCLA invokes the services of LTP. However, LTP relies on the data transmission services provided by the aggregated protocols below LTP in the stack, similarly identified as a single conceptual “*link service layer*.”

From the DTN perspective, RTTs in laboratory experiments are insignificant. For deployment in space flight missions, however, the protocols at the link service layer would potentially encompass interplanetary links for which RTTs may be on the order of minutes, hours, or even days.

Various CL protocols, TCPCL, UDPCL or LTPCL, can be used in the working environments of DTN. LTPCL has better throughput performance in long-delay environments than TCPCL [25] and UDPCL [26]. LTP is less affected by channel noise and delay than TCP; thus, it is a better solution than TCPCL for space missions [27]. DTN performance is affected by packet size; in general, bigger packet size results in better delivery performance [28]. However, in a noisy channel, a big bundle/block negatively impacts the performance due to the incidence of retransmission, as the possibility of packet corruption is higher for larger packets. Wang et al. [29] concluded that a higher number of smaller blocks would yield higher throughput performance than a lower number of large blocks because a larger block suffers more frequent packet corruption than a smaller one. Corruption of any segment of a block requires an additional retransmission cycle and costs as much as an additional increment of session closing time, equivalent to the duration of RTT [30].

IV. AGGRESSIVE AND PROACTIVE LTP CONTROL

Real-time operation of space missions by ground commands has to be treated as time-critical. It can be severely interrupted when these telecommands are not provided smoothly and promptly. When DTN is in full-fledged deployment in deep space missions, and when eventually spacecraft, ground network, and interplanetary network interconnect, spacecraft will be operated via the interconnected ground-space network, and all the interconnected network will run over DTN protocol.

Timely command reception and precise command reception completion time estimation will be essential for real-time operation. Unlike terrestrial networks, space communication is not as reliable as ground networks. Therefore, there is no guarantee of 100% timely, accurate information exchange for reliable spacecraft operation among communication assets in ground and space, not to mention data integrity, including command and telemetry delivery. Although LTP guarantees 100% data delivery, it allows out-of-order delivery and unbounded session closing time. Eventually, there is a

chance that some portions of spacecraft operation commands and telemetry packets will be delayed for a non-negligible amount of time; there should be tolerance for an allowable delay duration. Minimal time discrepancy would be preferable because tardy spacecraft operation hampers good QoS and the responsiveness of relevant systems.

DTN runs over BP and LTP, and LTP is responsible for reliable data delivery. To achieve a better QoS level using DTN and to control spacecraft as securely as possible in terms of realtime control, LTP session closing time must be bounded with minimal variations, which is the main objective of this study. We found that the unpredictability of LTP session closing time is closely related to the unpredictability of data segment (DS) loss and retransmission during an LTP transaction.

We propose a method that can efficiently aid the LTP segment control mechanism to finish the LTP session closing time in the shortest possible time. The method that we present requires some amount of redundant retransmission. However, we believe it provides better and more effective support for realtime operation in space environment over DTN fabric. In a lossy channel, the LTP session closing time highly depends on the OWLT and the time-out values for the checkpoint (CP) and RS retransmission timers [31]. OWLT is a physically given value and a parameter that cannot be altered by any means. The time-out value for the CP and RS retransmission timer is systematically determined for optimized performance. The time-out value should be greater than or equal to 2 times OWLT. During the assessment of our simulation results, we assume that all timeout values are perfectly set to optimal values so that the time overhead due to error in time-out values is negligible.

Various research studies noticed that communication between ground and spacecraft is vulnerable to intermittent connectivity loss, caused by antenna misalignment, attitude pointing error, weather conditions, etc. Considering the relatively long distance of deep space missions, an LTP session block's radiation time, bandwidth, and data rate are of negligible impact on data delivery latency compared to OWLT. The time required for recovery from any LTP segmentation loss event is at least 1 and 1/2 RTT ($= 3 \times \text{OWLT}$), as shown in Fig. 1, assuming (a) the LTP segment signaling a transmission "CP" has not been lost, and (b) the first retransmission of the lost segment is delivered successfully.

The loss of the last LTP segment causes failure of the control signal delivery necessary to elicit an appropriate action from a receiver. Therefore, a sender has to wait for another RTT (a CP timer expiration event) to catch the situation, adding at least one RTT penalty for the LTP session closing because a time-out period is supposed to somewhat greater than one RTT. The severity of the additional latency in session delivery increases as the OWLT increases. Considering the RTT between ground and Moon as 2.56 s, even a single loss event of the last DS of an LTP session transaction which has a CP can result in at least 7.68 s (6×1.28 s) or more additional latency of completion of the delivery of all segments to a remote receiver at a cislunar distance even when there is no further segment loss expected. More complicated segment corruption cases can happen, as

shown in Fig. 2.

In a system that requires time-sensitive or real-time operation in ground or deep space, the uncertainty of the session completion time can introduce catastrophic results, and this unpredictable latency should be avoided or minimized as far as possible.

To reduce LTP session closing latency variation, we present an aggressive and proactive control mechanism for CP signal management. The CP signal is a key control soliciting feedback from a receiver engine. The core idea of aggressive and proactive CP (APCP) signal management is to let a receiver engine know the approximate size of an LTP session block. As a result, the receiver engine knows the approximate anticipated arrival time of the CP signal. Therefore, it can issue a discretionary RS even if the last DS possessing a CP signal is lost during transmission. The sender engine will receive the discretionary RS from the receiver engine instead of waiting for time-out during the CP retransmission timer, which generally costs one RTT. Therefore, this scheme can save one RTT penalty if the lost CP is successfully delivered.

Various ways of enabling this scheme are possible. However, we would like to claim that LTP session closing time can be optimized most effectively by leveraging the following capabilities:

- Capability of inferring LTP block size prior to reception of CP
- Capability of reducing the probability of retransmission failure
- Capability of discretionary control signal self-generation

Some methods require a modification to the current LTP specification, while others do not. We present some reflections on these concepts and approaches in the following sections. In this section, our primary contribution is a detailed analysis of the self-management capability of generating discretionary control signals.

A. Erasure Coding for Reduction of LTP Segmentation Retransmission

Before discussing our proposed technique, it is necessary to discuss the role of erasure coding (EC) in deep space communication. EC can reduce LTP session closing time latency because EC brings the BER in a communication channel close to the theoretically minimal value [32]. In a very clean communication channel with minimal BER, the probability of retransmission of LTP segments becomes low. Because EC requires extra redundant bits for holding parity information, an LTP transaction that uses EC has a performance penalty in terms of the overall rate of goodput. We performed an analysis to compare the performance difference between two EC schemes: Reed-Solomon and LDPC. Reed-Solomon provides moderate performance improvement with moderate overhead, and LDPC provides ideal channel coding performance with more severe overhead. Although Reed-Solomon yields a higher chance of retransmission due to a higher probability of channel error, it has higher benefits in terms of goodput than LDPC.

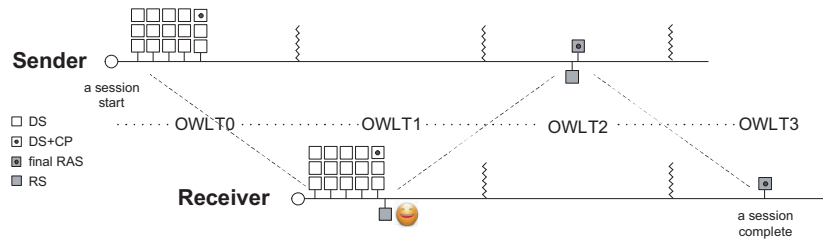


Fig. 1. The perfect and shortest LTP session closing scenario with channel error free condition.

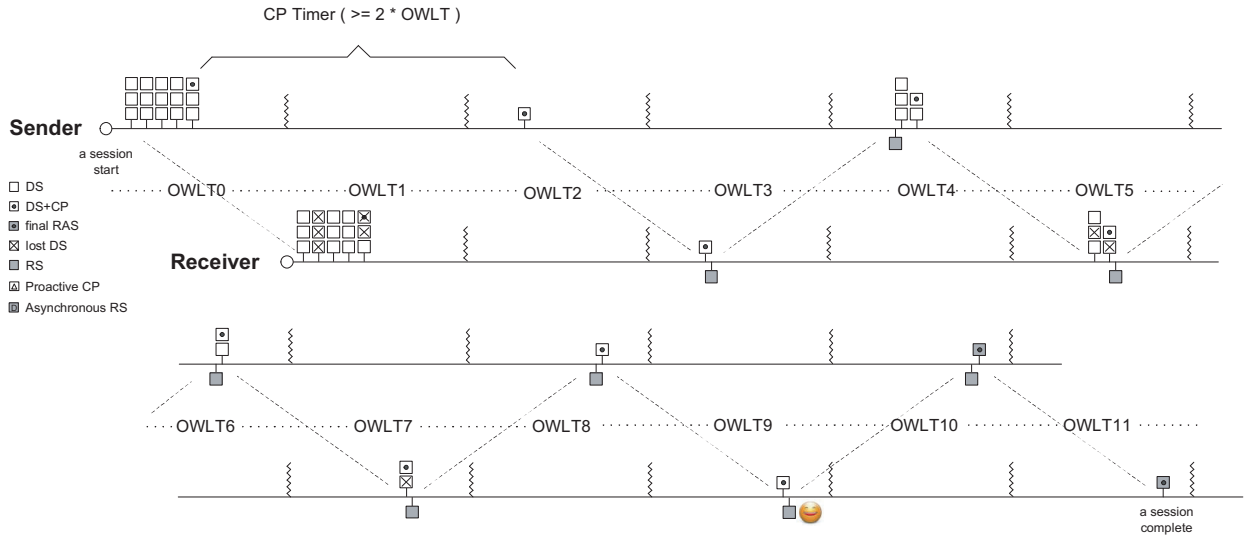


Fig. 2. Worse LTP session closing scenario when severely suffering from packet failure occasions.

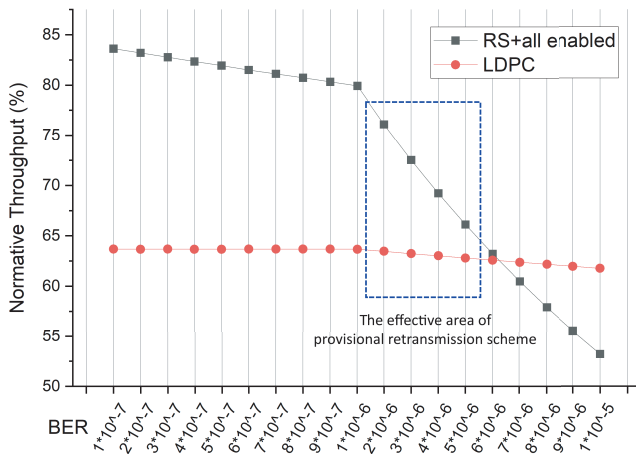


Fig. 3. Comparison results of RS+all enabled configuration and LDPC condition.

When APCP and provisional DS retransmission (PDR) are used, goodput rate declines. However, using Reed-Solomon and all proactive measures (APCP and PDR) together has been shown to be still more beneficial under some circumstances (refer to the blue rectangle), as depicted in Fig. 3. Adapting LDPC EC is even more effective than Reed-Solomon, especially when APCP and PDR are additionally

TABLE II
Inferring receiving segmentation status.

Interim segment	CP Inferred region	Expected CP sequence number
1st	1/4 of total block	1 to $2^{12} - 1$
2nd	1/2 of total block	2^{12} to $2^{13} - 1$
3rd	3/4 of total block	2^{13} to $(2^{13} + 2^{12}) - 1$

enabled in a very low BER environment.

PDR is less effective in a better BER channel (right side of the blue rectangle). However, the proposed scheme is still worth applying since the low possibility of packet corruption due to low BER results in a low probability of retransmission, thus in low overhead. When the BER is higher than 6×10^{-6} , LDPC EC is more efficient because it has better goodput and lower retransmission possibility. On the other hand, choosing Reed-Solomon in a bad channel link environment results in higher packet corruption, thus more frequent retransmissions and a lower goodput rate than is provided by LDPC.

B. LTP Block Size Inference Service with Proactive CPs and Proactive RSs

According to the current LTP specification, the LTP block size can only be known to a receiver if a CP with end of block (EOB) at the last DS block is successfully delivered. If the CP were missed, the receiver could not determine whether

TABLE III

Considerable points on the CP handling mechanism of APCP.

Principles & Assumptions
<ul style="list-style-type: none"> • A receiver has no obligation to respond to the incremental CPs. • A receiver can estimate the size of the current session, so it can issue a discretionary RS even final DS+CP segment s lost during transmission. • A sender may decline or ignore an RS issued by its incremental CP. • A receiver must respond to the last DS+CP, so it must issue an RS according to the CP. • A sender must respond to the RS for the last CP, so they must transmit the designated lost DS again.

the current block is being closed or the channel is suffering some service failure. The problem is that at least two OWLT, i.e., one RTT, must elapse before corrective action can be taken at the sender and receiver. To mitigate this aspect, we have devised a way of transmitting three discretionary CPs prior to the final CP at the EOB to avoid unnecessary ambiguity. This concept does not conform to the standard LTP specification; we would propose a small revision to the standard.

We contemplate two possible mechanisms that enable the receiver of an LTP block to infer the approximate size of that block long before the arrival of the EOB segment; one of these mechanisms entails a revision of the LTP specification, and the other does not. In both cases, the inferred block size is guaranteed to be no less than the actual size of the block.

1) *The first method will not require specification change:*
In conformance to a service-level agreement (or equivalent) between the sending and receiving engines:

- The sender sets the CP flag on three DSs that precede the EOB segment as shown in Table II.
- The receiver can compute an estimate of the block's total size from the reception of any one of these three CP segments: the value of the CP sequence number indicates whether the CP segment's offset is approximately 1/4, 1/2, or 3/4 of the block size.

Interim CPs and responding RSs are defined in the current LTP Blue Book. This mechanism imposes up to 6 additional octets of overhead per transmitted LTP block for which this feature is enabled. In the unlikely event that all three interim CP segments are lost, the estimate of the total block size cannot be computed.

2) *The second method will require specification change:*
The version number field in the LTP segment header is reduced in length from 4 bits to 2 bits, increasing the length of the segment type flags field from 4 bits to 6 bits. One of these two additional flag bits is left undefined for future use; for DSs only, the other additional flag bit is used to indicate the presence or absence of a "block size indication octet" in the segment's header. When this octet is present, its low-order 7 bits indicate the fraction of the total block size represented by the DS's ending offset (the sum of the DS's offset and length), expressed in 128ths. (For example, a value of 0010011_b = 19 indicates that this segment's ending offset is about 19/128 of the total length of the block, about 15%.) The first bit is set to 1 – and the other 7 bits are set to 0 – if the ending offset of

this segment is less than 1/128 of the total length of the block. If the total length of the block is unknown to the sender, then all 8 bits of the octet are set to 0.

The receiver can compute an estimate of the block's total size from the reception of the first segment whose block size indication octet is neither 1000000_b, nor 00000000_b: The approximate block size is obtained by dividing the segment's ending offset by the indicated fraction.

This mechanism imposes one additional octet of overhead per transmitted LTP DS for which this feature is enabled. Assuming the block size is known to the sender, the first received segment will, at minimum, provide a floor for the block size: If the first bit of its block size indication octet is 1, then the size of the block is known to be more than 128 times the segment's ending offset. For any block that is transmitted in no more than 128 segments, the approximate size of the block is computed by dividing the ending offset of the first received segment by the indicated fraction.

To enable the above mechanisms, invoking voluntary or discretionary LTP control segments (CP and RS) without waiting for certain requests or time-out expiration events must be allowed. We believe that this mechanism helps respond more effectively to small transmission failures that potentially can lead to huge transaction delays. Table III presents some considerations for the CP handling mechanism of APCP.

C. Provisional Retransmission of DS Requested by RS

Unless the operating environment of DTN is channel error-free, segmentation corruption and delivery failure cannot be avoided. The rate of LTP segmentation delivery failure depends on the BER on the communication channel of the operating environment. BER can be enhanced by applying EC, but attempting to establish an operating environment of error-free communication is not attractive for most space missions due to the transmission overhead of EC.

If data loss cannot be prevented then any DS that is not delivered must be retransmitted. It costs at least one RTT for a sender engine to notice the successful reception of the retransmitted DSs at a receiver LTP engine. To reduce LTP session delivery latency as short as possible, which is one of the main objectives of this study, repeated delivery failure of the retransmitted DSs must be avoided at all costs because this event incurs another RTT penalty. From a system perspective, the severity of the time penalty becomes increasingly significant as RTT becomes longer in deep space missions.

To increase the chance of successful reception for the retransmitted DSs, we propose provisional retransmission for all DSs requested by the RS. Because we do not know which segment(s) will be lost again, all of the retransmitted DSs will be reretransmitted as a "Booster shot." According to our calculation, the possibility that a further retransmission request will be issued despite this provisional retransmission is very low. Therefore, almost all LTP transactions will be completed within two RTTs.

This scheme can be realized without modification of the current LTP specification. A conformant receiving LTP engine must be able to handle duplicated arrival of DSs and possibly

duplicated CP serial numbers in segments received from a given sender LTP engine. We propose half of one OWLT – but at least one second and at most five seconds – as the interval between original retransmission and provisional retransmission. However, the selection of this value must be mission-specific choice and can be determined by considering BER, current communication link status, current spacecraft position, and so on. Except for rare cases in which the two consecutive retransmissions still fail to deliver all initially lost segment, a receiver LTP engine will save one RTT when some of the original packets are corrupted and retransmission is required to recover them.

This scheme will not work when the link is temporarily unavailable due to loss of line of sight, communication hardware failure, or service postponement. This scheme is only effective at segment loss resulting from independent or burst bit corruption while signal radiation and propagation continue without interruption.

The reduced LTP session closing time enables the earliest possible release of resources allocated for the affected LTP session, improving system resource management.

D. Friendly Generated RS

A receiver engine must positively acknowledge all transmitted DSs to finish the current session's transaction from the sender engine's side. So, if some of the RSs acknowledging successful delivery of DSs are lost during the transaction, the closing of the current session will be delayed until expiration of the report ACK timer causes the receiver engine to retransmit the report. This can happen when a receiver engine transmits fragmentary reports in response to fragmentary CP signals from a sender engine. By the current LTP specification, a receiver engine is not obligated to include in its reports any portions of the block that are outside of the offset and length of a received CP. Hypothetically, in a highly noisy channel, a full positive report claiming all the areas of a session could be very fragmented. Any missed fragmentary report would lead to non-negligible latency of session closing time when the channel is lengthy.

Therefore, we want to propose a “friendly” RS concept. When a receiver engine receives all DSs successfully, it is recommended that the receiver send a fully positive acknowledged RS whenever the CP requests it from a sender engine, even if it is not required to be fully acknowledged for any portions of the data block outside of the offset and length information stated in the CP. This will enable a sender engine to close the current session as soon as possible whenever at least one fully claimed RS is received.

This concept is especially beneficial when DSs' aggressive and provisional retransmission is applied. The sending engine sends the DSs to be provisionally retransmitted consecutively with a little delay. A friendly RS, which claims all DSs even if a requested CP does not specify the entire reported scope, can mitigate delay in ACK of the original and provisional retransmission DSs.

This scheme boosts the effectiveness of proactive and provisional retransmission. Two replying reports will be issued, and

there is a non-zero probability that one of the reports will fail. The friendly RS can mitigate segment delivery failures that may be recoverable when the first report is lost among the duplicate transmission. LTP session closing time is reduced with a minimal penalty.

Finally, the friendly RS minimizes the complexity of handling ACK of 100% successful reception at a receiver engine. Report processing does not require complex computation because there is no need for searching multiple reception claim information structures to find unfilled holes in the sequence of DSs.

V. SIMULATIONS

We developed simulation software for this study and performed simulations to prove that the proposed mechanisms effectively expedite LTP session closing. All the issues mentioned above were considered as simulation factors during the simulation design. The main purpose of the simulation software was to provide a simulated profile of LTP session closing times, ordered by OWLT, according to selected simulation parameter values. Goodput is also reported. It is worth noting that this simulation software does not implement LTP operation and functionality itself, but only space communication conditions and their effects on the timing behavior of LTP transactions. While the authors have LTP implementations which have been functionally demonstrated in previous studies [33], [34], only the timing functionality for beginning and ending LTP sessions was required in order to study the LTP session closing mechanism and only the temporal behavior model of LTP session delivery service had to be considered in the simulation.

A. Design of Simulation Workbench

The simulation software was developed to run in an environment characterized by the following inputs:

- Various BERs used to cause segmentation loss, 10^{-5} , 5×10^{-6} , 10^{-6} , 5×10^{-7} , and 10^{-7}
- Original LTP control algorithm vs. ACP and PDR
- Various segmentation sizes, used to compute goodput rate for real data only, 1k, 2k, and 3k ($k=1000$)
- Simulation runs (200,000 OWLT rounds)

By the LTP specification document, RFC-5326 and related work [35], LTP session closing time is calculated by (1).

$$\text{LTP session closing time} = (1 + 2N) \cdot \text{OWLT}, \quad (1)$$

where N is the required number of segmentation negotiation runs necessary for closing an LTP session in perfect condition; for an error-free channel, N is 1.

We consider that an LTP session is effectively closed when a receiver receives a report ACK from a sender for a receiver's full claim report. In general, the determination that an LTP session is closed in what circumstances is an implementation matter. For example, a receiver can close its LTP session when it receives full DSs or when it receives the report ACK from a sender; the earlier determination is shorter by one RTT. However, for this study we prefer waiting for a report

ACK from the sender before closing the current LTP session because even if a receiver receives all the DS 100%, there is a chance that a sender thinks the current session is a failure and terminates after several retries by retransmission time-out events. Of course, it would be a very low probability, but conservatively, we are more confident of having closed the current LTP session when 100% safe conditions are met.

We assume that the produced report and report acknowledge segments are radiated once they are created to eliminate this potential time delay from simulation results. However, the space link channel is generally busy, so some preparation time for radiation cannot be avoided in practice. One technique to reduce this delay is to give higher priority to the report and report acknowledge segments during the process of enqueueing them to a radiation transmission channel.

Note that simulation results will not depend on how long OWLT is, as depicted in (1), which states that LTP session closing time is always proportional to OWLT by positive integers, e.g., 3, 5, 7, etc. However, OWLT in all cases is effectively long enough to hold a sufficient number of LTP sessions initiated during simulation time. In this simulation, we choose 3.6 seconds as the OWLT parameter value, approximately 1.08×10^6 km distance from Earth. Also, communication bandwidth does not create any difference in terms of LTP session closing time and relative throughput analysis during simulation. Bandwidth was required to be large enough to populate a sufficient number of LTP transaction cases. The time for preparing a segment and reading it is ignored, which means all input/output buffer is ideally balanced so that no lapse of time in segment data handling is involved.

In this simulation, only the receiver engine's temporal behavior is monitored and analyzed because the receiver side always ends with a longer completion time for waiting for report ACK from the sender's side. Critical simulation parameters identified in related work [36] were applied during the simulation. Table IV presents major simulation parameter values for the simulation.

During simulation, LTP session closing events were recorded at a memory array for OWLT slots 1, 3, 5, 7, 9, 11, 13, and 15. For channel error-free conditions, all sessions end at 3 OWLTs theoretically. Any transmission failure during the transaction imposes one RTT ($2 \times$ OWLT) latency penalty at the receiver.

The determination of packet error occurrence is based on the BER and probabilistic decisions by pseudo-random generation. We use (2) for calculating the segment loss (error) rate, SegLossRate [37],

$$\text{SegLossRate} = (1 - (1 - \text{BER})^{\text{SegSize}}) \times 1,000,000 \quad (2)$$

to determine a segment loss event according to the segment loss rate on segment data, a pseudo-random number between 1 and 1,000,000 is generated. If the random number is between 1 and the segment loss rate calculated by (2), delivery of the segment is deemed to have failed. For example, considering BER as 10^{-5} and 1000 bytes of segment length,

$$\begin{aligned} \text{Segment loss rate} &= (1 - (1 - 10^{-5})^{1000 \times 8}) \times 1,000,000 \\ &= 76,884. \end{aligned}$$

This segment has 7.6884% of segment loss rate when it transfers to a destination via a channel of 10^{-5} BER. If a randomly generated value from 1 to 1,000,000 hits one value between 1 and 76,884, this segment can be deemed a delivery failure.

We consider the simulation conditions as follows:

- 1) Segment delivery failure accounting model by randomly generated session information, e.g., session length;
- 2) Consider only LTP's red data, as green data does not require retransmission;
- 3) Configurable simulation parameters intended: Segmentation error rate, probability of DS without CP loss, probability of DS with CP loss, probability of control segment (RS, RAS) loss, segmentation length, count of segmentation per session (randomly chosen), RTT, channel coding overhead, space link protocol overhead;
- 4) Expected outputs – rounds that are required to complete a session, normally one RTT per round, according to various conditions;
- 5) We did not count overhead from the space link format frame because it is user dependent and can mislead the effect of the way we present. However, the overhead from the user case should be easily computed from the actual used space link layer protocol chosen by the user;
- 6) We did not consider performance delay; retrieval from sending buffer space is assumed to be instantaneous, and there is no wasted time in transmission. A receiver can respond immediately without delay.
- 7) BER is the post-FEC, i.e., no FEC will be considered; the system link budget design can choose the FEC method. For example, KARI's heritage is Reed-Solomon (255, 223, interleaved) for TM, NASA's recommendation is LDPC (7, 1/2) in deep space;
- 8) The size of the LTP header is not considered in goodput analysis as it is a small fraction of segment size [28].

We consider some assumptions for goodput computation as follows:

- 1) LTP format overhead is not accounted for except CP and RAS. Minor overhead was not included in the goodput computation;
- 2) All retransmitted DSs are not included in the goodput estimation;
- 3) All transmitted CP and RAS are not included in the goodput estimation;
- 4) CCSDS TC/TM/AOS transfer frame overhead is not accounted;
- 5) If necessary, RS coding overhead is estimated by 1.14, for LDPC 1.5;
- 6) It is assumed that each segmentation is instantly enqueued at the channel link buffer, so there is no delay during segmentation packets processing.

B. Simulation Results

We run each simulation 200,000 OWLTs per condition. Tables V and VI present simulation results for goodput per simulation conditions and applied LTP session handling methods.

TABLE IV
Major simulation parameters.

Simulation parameters	Unit	Value & Equation in C expression
owlt	Micro-second	3600 * 1000
ber	Bit-error rate, 1/BER	#define BER_10_6_1 1000000 /*1 * 10 ⁻⁶ */ #define BER_10_6_2 500000 /*2 * 10 ⁻⁶ */ #define BER_10_6_3 333333 /*3 * 10 ⁻⁶ */ #define BER_10_6_4 250000 /*4 * 10 ⁻⁶ */ #define BER_10_6_5 200000 /*5 * 10 ⁻⁶ */
seglossrate	Segment loss rate over 1,000,000 ^a	Calculated by (2)
ssize	Segment size per transmission, bit	1000 * 8, 2000 * 8, 3000 * 8
bw	Bandwidth, Mbps	20
rascost	RAS cost, micro-second	10
cptimer	CP retransmission timer, micro-second	OWLT*2
rstimer	RS retransmission timer, micro-second	OWLT*2
ctrlretrcost	Cost of ctrl segment retransmission, micro-seconds	100
ctrlseglength	Length of ctrl segment (RS, RAS), byte	80
maxsize	Max bundle size, kbyte	550
maxsimcount	Max simulation time, OWLT	200,000

^aIf a seed value is 10,000, loss rate is 1%, and if a seed value is 100, loss rate is 0.01% by considering that pseudo-random generation numbers are evenly distributed in 1 and 1,000,000.

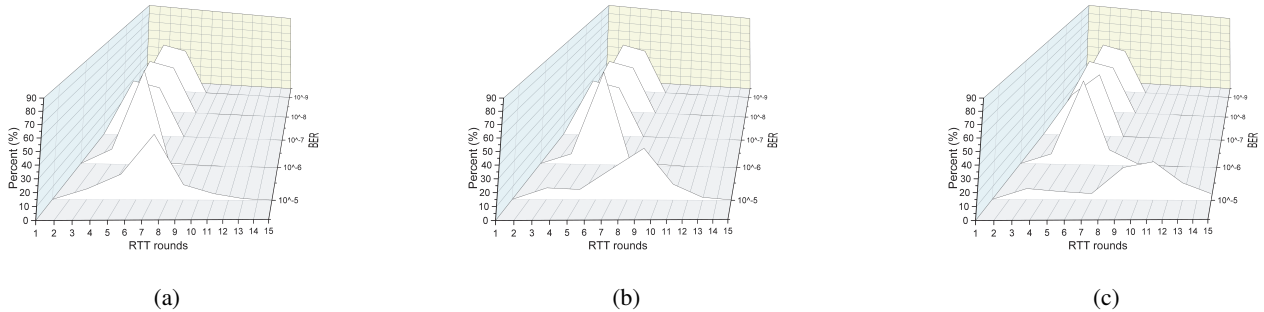


Fig. 4. Nominal behavior (a) 1k segment size, (b) 2k, and (c) 3k.

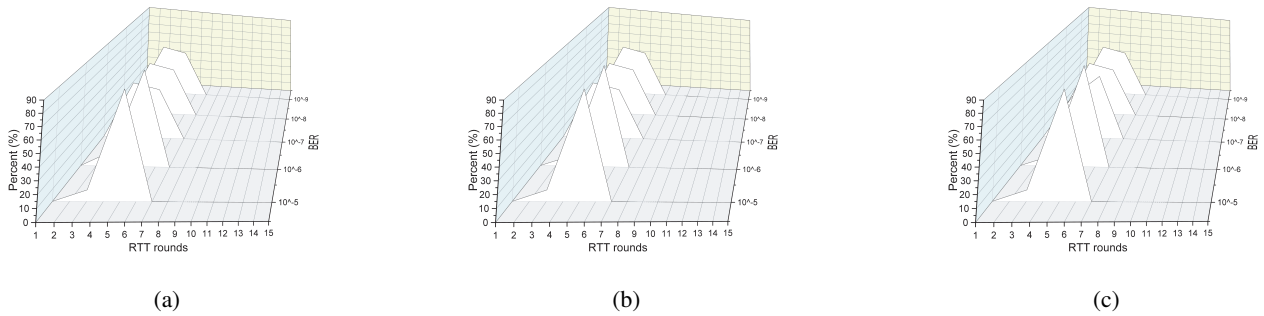


Fig. 5. Aggressive & proactive CP and provisional retransmission (a) 1k segment size, (b) 2k, and (c) 3k.

These results indicate that APCP has no effect on goodput performance. Provisional retransmission shows slightly lower goodput performance because it retransmits DS as redundancy, and the volume of the retransmitted DS becomes bigger when BER increases. When BER is very low, the goodput penalty from the provisional retransmission is small, but it has a bigger advantage regarding the required OWLT rounds for LTP session closing.

Table V shows goodput comparison results with various BER and segment sizes for each method. The differences in goodput results are shown in Table VI. The normal (no

changes from LTP specification) method is a datum point of the comparison results. APCP and provisional method show lower goodput, and the severity of the goodput drop is increases as segment size and BER increase. When 5×10^{-7} BER becomes a datum point of the applicable BER ranges, however, the loss of goodput due to the most provisional method is marginally negligible as -1.291% compared with the benefit from the bounded and short session closing time.

Fig. 4 shows a simulation result of analyzing RTT rounds distribution varying BERs and segment size in which simulation time is as long as 200,000 OWLTs. Because a segment

TABLE V
Goodput comparison (Throughput, %).

Method	Segment size	BER				
		10^{-5}	5×10^{-6}	10^{-6}	5×10^{-7}	10^{-7}
Normal	1k	83.616608	90.338425	96.288300	97.096703	97.567902
	2k	69.256546	81.149399	92.648460	94.205254	95.523483
	3k	56.354717	72.256256	88.904305	91.270851	93.254112
APCP	1k	83.630920	90.353958	96.294083	97.094131	97.566811
	2k	69.285767	81.172554	92.637726	94.207115	95.518097
	3k	56.378498	72.290489	88.901871	91.299721	93.257027
APCP+PDR	1k	77.952217	87.008652	95.504745	96.696342	97.392609
	2k	60.645611	75.392601	91.102623	93.394554	95.336578
	3k	46.231937	64.645248	86.673851	90.092621	93.002434

TABLE VI
Goodput comparison (Throughput difference, %).

Method	Segment size	BER				
		10^{-5}	5×10^{-6}	10^{-6}	5×10^{-7}	10^{-7}
Normal	1k	0	0	0	0	0
	2k	0	0	0	0	0
	3k	0	0	0	0	0
APCP	1k	0.017	0.017	0.006	-0.003	-0.001
	2k	0.042	0.029	-0.012	0.002	-0.006
	3k	0.042	0.047	-0.003	0.032	0.003
APCP+PDR	1k	-6.774	-3.686	-0.814	-0.412	-0.18
	2k	-12.433	-7.094	-1.668	-0.861	-0.196
	3k	-17.963	-10.533	-2.509	-1.291	-0.27

having a larger size is more vulnerable to checksum failure than a smaller size, the 3k segment size case shows a very uncertain LTP session closing. The large segment block has to be avoided when bad BER is expected. However, if BER is better than 10^{-7} , all segment size cases show uniform LTP session closing and can be concluded as bounded.

In a high BER, 10^{-5} and 3k segmentation size, cases with intolerable delays have been observed. Over 50% of transactions are delivered after the 11th OWLT or later. On the other hand, in BER under 10^{-7} , over 99% of transactions are completed no later than the 5th OWLT. It should be noted that the 3rd OWLT is the ideal case under error-free conditions.

Fig. 5 shows a simulation result with the same conditions applied to Fig. 4 but applying APCP and PDR. The results show enhanced and bounded LTP session closing time regardless of segment size variations. 3k segment size block still shows bounded LTP session closing at lower BER than 10^{-6} . In realtime deep space applications, bounded transaction completion time is more valuable despite some loss of goodput if it is marginally acceptable.

Tables VII–XI summarize simulation results with variations of BER, segment size, and applied LTP session closing method.

Fig. 6 provides an assortment of simulation results as charts with various conditions applied during the simulation. Fig. 6 presents comparison results on 1, 2, and 3k segment sizes with BER variations between 10^{-5} and 10^{-7} each method is applied; a sequence of nominal operation without applying any of APCP or PDR, a sequence of aggressive CP (aggCp)

for APCP only, and a sequence of proactive DS (proDs) for APCP+PDR. Upper charts show the results when nominal LTP operation is applied, and lower charts show the results when the PDR scheme is applied. Except for very fine BER, 10^{-7} , the PDR scheme improves BER, similar to channel coding, but with much less throughput penalty. For instance, Reed-Solomon coding Reed-Solomon (255, 223) constantly consumes an FEC ratio of 1.14 (14% overhead). PDR scheme incurs 2.5% overhead as some corrupted LTP segments are retransmitted, not the whole LTP block.

Fig. 7 presents comparison results on 1, 2, and 3k segment size with BER variations between 10^{-5} and 10^{-7} with nominal and all APCP+PDR enabled; upper charts for nominal, and lower charts for all APCP+PDR enabled. As can be seen in Fig. 7, a ceiling BER of 5×10^{-6} is a major effective range obtaining the bounded session closing time thanks to APCP+PDR.

Fig. 8 shows a simulation result performed for BER conditions of concern. BERs between 10^{-5} and 10^{-7} are common environments for spacecraft in deep space encounters. From 5×10^{-6} and 10^{-5} in nominal operation in LTP, cases for LTP session closing time longer than 7 OWLTs rounds begin to be observed. When all APCP and PDR are enabled, almost all sessions are closed within 5 OWLTs rounds.

Fig. 9 presents more detailed observations for BER conditions of 5×10^{-6} and 10^{-6} . When all mechanisms are enabled, the shape of session closing seems very close to the one of 5×10^{-7} BER, which means that the proposed scheme contributes to the effective reduction of BER.

TABLE VII
Required OWLT rounds under 10^{-5} BER.

Method	Segment size	Percentage(%) of required OWLT rounds under 10^{-5}							
		1	3	5	7	9	11	13	15
Normal	1k	0	8.286015	21.63052	52.564633	11.308482	4.922283	1.121612	0.148454
	2k	0	7.660693	7.189646	27.802929	34.813723	12.619294	6.949297	2.278206
	3k	0	7.045591	5.595466	7.985961	30.391005	23.719002	13.188756	7.648744
APCP	1k	0	8.30054	23.370753	56.835473	10.517384	0.900809	0.069188	0.005241
	2k	0	8.051538	8.368188	32.067803	39.609751	9.98248	1.62939	0.249568
	3k	0	7.830897	7.022279	9.081697	36.890444	28.11874	8.519165	1.991697
APCP+PDR	1k	0	8.326922	90.642571	0.931085	0.091503	0.007196	0.000635	0.000088
	2k	0	8.018009	88.643754	3.051202	0.252302	0.029357	0.004501	0.000757
	3k	0	7.851314	78.254783	11.763634	1.834401	0.253363	0.035587	0.006053

TABLE VIII
Required OWLT rounds under 5×10^{-6} BER.

Method	Segment size	Percentage(%) of required OWLT rounds under 5×10^{-6}							
		1	3	5	7	9	11	13	15
Normal	1k	0	8.642193	56.649578	29.564685	3.747247	1.27981	0.110142	0.006013
	2k	0	8.320622	21.658717	52.609974	11.252815	4.897453	1.095636	0.147253
	3k	0	7.969341	10.197448	44.679126	24.189307	8.44658	3.585577	0.777713
APCP	1k	0	8.531938	58.941472	30.837557	1.61879	0.067268	0.002848	0.000127
	2k	0	8.314598	23.382021	56.884575	10.441265	0.899288	0.072117	0.005773
	3k	0	8.207932	11.475662	49.968725	26.023608	3.821768	0.446266	0.050195
APCP+PDR	1k	0	8.554997	90.852493	0.549352	0.041443	0.00165	0.000066	0
	2k	0	8.305114	90.765762	0.831339	0.090036	0.007111	0.000531	0.000106
	3k	0	8.18501	90.224224	1.424843	0.145687	0.017743	0.001984	0.000454

TABLE IX
Required OWLT rounds under 10^{-6} BER.

Method	Segment size	Percentage(%) of required OWLT rounds under 10^{-6}							
		1	3	5	7	9	11	13	15
Normal	1k	0	13.012162	84.441268	1.837084	0.689564	0.019585	0.00034	0
	2k	0	8.854067	83.098656	6.479984	1.438977	0.124583	0.00364	0.000093
	3k	0	8.765545	75.348741	13.385916	2.123913	0.360454	0.014772	0.000659
APCP	1k	0	13.000317	85.087651	1.891522	0.020186	0.000324	0	0
	2k	0	8.698769	84.452188	6.722549	0.12413	0.00227	0.000093	0
	3k	0	8.647359	77.172863	13.791712	0.377527	0.010305	0.000235	0
APCP+PDR	1k	0	12.978132	86.848503	0.16709	0.006209	0.000062	0	0
	2k	0	8.740481	90.935034	0.308565	0.015765	0.000158	0	0
	3k	0	8.661182	90.960526	0.354347	0.023604	0.000337	0	0

TABLE X
Required OWLT rounds under 5×10^{-7} BER.

Method	Segment size	Percentage(%) of required OWLT rounds under 5×10^{-7}							
		1	3	5	7	9	11	13	15
Normal	1k	0	25.059521	74.092865	0.542571	0.301876	0.003135	0.000031	0
	2k	0	13.043511	84.41835	1.825532	0.693749	0.018611	0.000247	0
	3k	0	8.865774	86.182874	3.821464	1.071223	0.057365	0.001256	0.000047
APCP	1k	0	25.065371	74.407381	0.522253	0.004919	0.000077	0	0
	2k	0	13.013382	85.092342	1.873758	0.020331	0.000185	0	0
	3k	0	8.79238	87.165642	3.983133	0.058009	0.000837	0	0
APCP+PDR	1k	0	25.063041	74.851352	0.08258	0.002994	0.000031	0	0
	2k	0	13.039264	86.798108	0.156906	0.005723	0	0	0
	3k	0	8.770191	90.989661	0.228614	0.011348	0.000188	0	0

TABLE XI
Required OWLT rounds under 10^{-7} BER.

Method	Segment size	Percentage(%) of required OWLT rounds under 10^{-7}							
		1	3	5	7	9	11	13	15
Normal	1k	0	52.629125	47.250748	0.081629	0.038437	0.000061	0	0
	2k	0	52.614254	47.146347	0.164208	0.074852	0.000337	0	0
	3k	0	39.429703	60.155135	0.263492	0.150975	0.000691	0	0
APCP	1k	0	52.670765	47.274369	0.054409	0.00046	0	0	0
	2k	0	52.689302	47.206417	0.102962	0.001318	0	0	0
	3k	0	39.485091	60.293365	0.219611	0.001934	0	0	0
APCP+PDR	1k	0	52.665538	47.319004	0.01498	0.000476	0	0	0
	2k	0	52.602845	47.36656	0.029641	0.000953	0	0	0
	3k	0	39.502499	60.450834	0.044957	0.00171	0	0	0

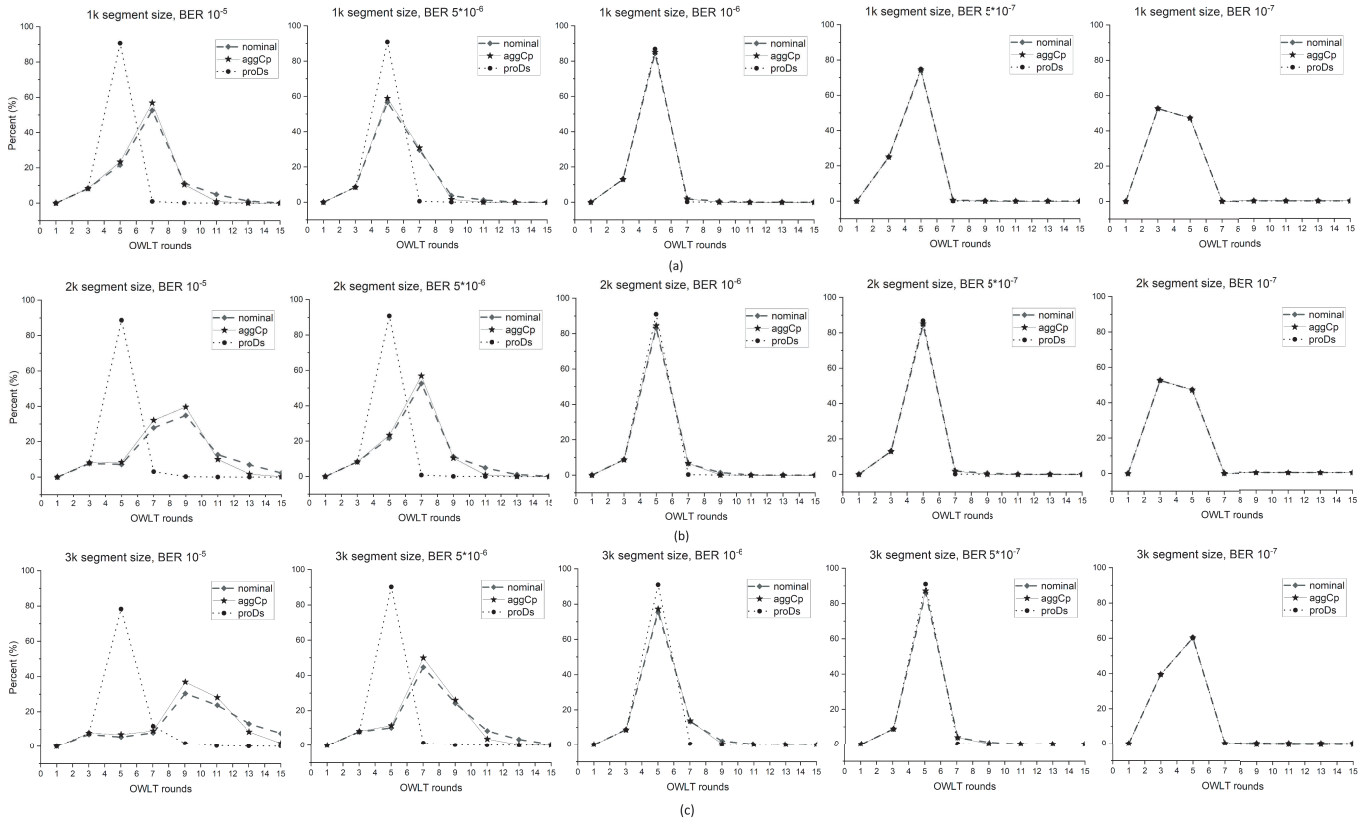


Fig. 6. Comparison of 1k, 2k, and 3k segment sizes under various BER and options.

Fig. 10 presents the same approach to a simulation done in Fig. 9, but with different BER conditions, 5×10^{-7} and 10^{-7} . In relatively clean channels lower than 5×10^{-7} , the proposed scheme does not provide any noticeable changes.

C. Analysis of the Simulation Results

The simulation results show that the aggressive checkpointing and PDR can significantly lower the probability of closing a session only after a large number of OWLT rounds, such as 7 OWLT rounds or more, by a factor of 10 compared to omission of these mechanisms. The benefit comes with a non-negligible goodput penalty when BER is high, e.g., higher than 10^{-6} , but it seems to be a tolerable cost in low BER like 5×10^{-7} or lower BER.

When APCP and PDR are enabled, temporal responsiveness of the LTP session closing has been further improved at 10^{-5} BER and 2k/3k segment size, and it still shows meaningful improvements at 10^{-6} BER and 2k/3k segment size. There are no significant improvements at BERs lower than 10^{-7} regardless of 1k/2k/3k segment size. Because the probability of segmentation delivery failure in a clean channel is very low, naturally there is no improvement and no throughput penalty from the aggressive and proactive retransmission scheme.

When APCP and PDR are enabled, it is observed that 98.97–99.99% of LTP sessions are closed within 3 or 5 OWLT in all BER configurations. In particular, in conditions of 2k segment size and 10^{-6} BER 99.67% of LTP session closing time are bounded within 5 OWLTs; the improvement provided by enabling this scheme amounts to a 8.39% shorter session

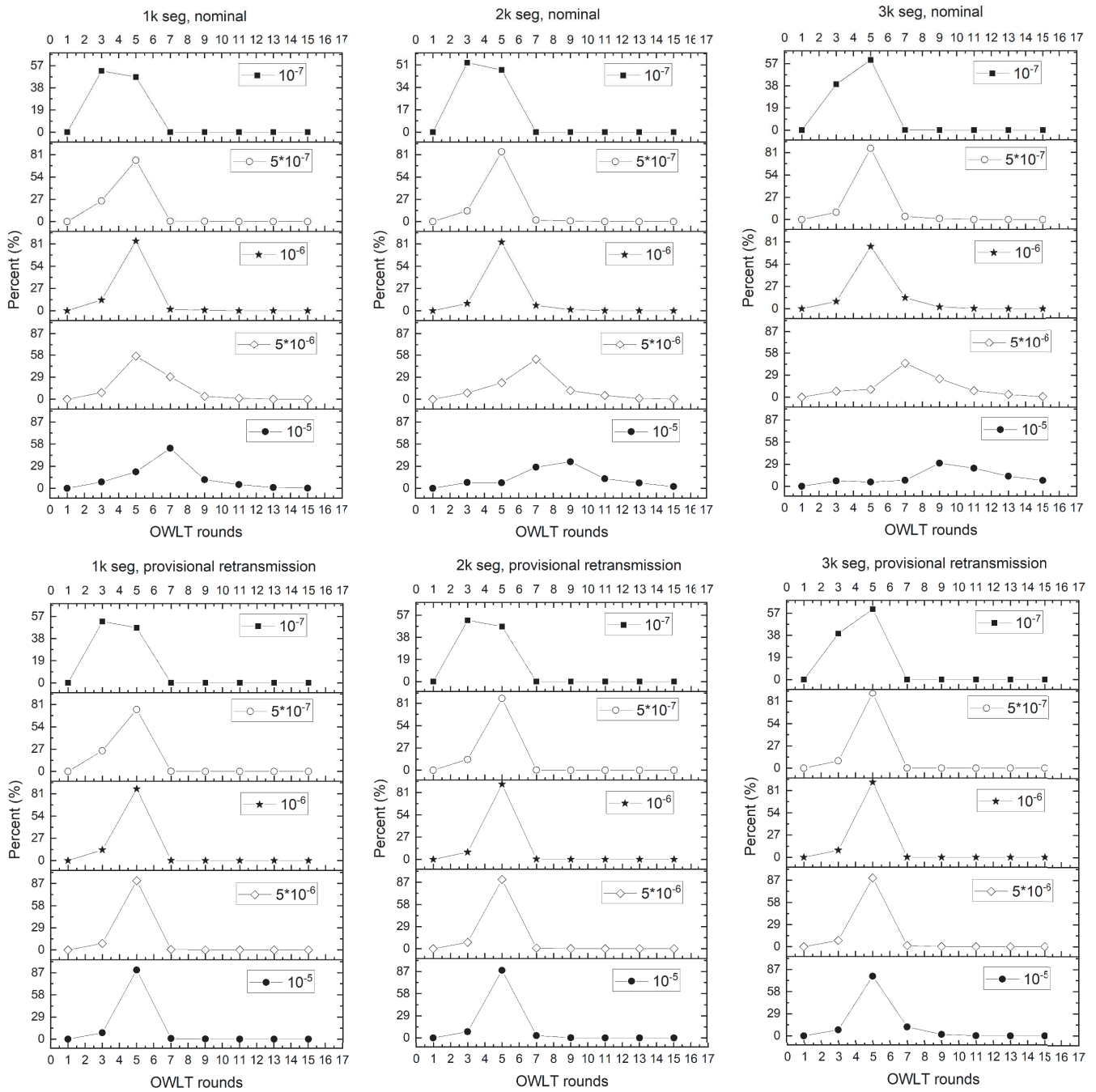


Fig. 7. Comparison of 1, 2, and 3k segment size with nominal and APCP+PDR.

closing time and only 4% chances of exceeding 5 OWLTs. There is still approximately a 1.03% chances of session closing requiring more than 7 OWLTs. If this scheme is not applied this probability rises to 2.45% at 10^{-6} BER and 3k segment size; session closure may take as much as 9–15 OWLTs, which could constitute a mission-critical situation.

By observing the results from the simulation shown in Figs. 8, 9, and 10, we conclude that there could be an optimal LTP segmentation size and applicable BER value on a transmission channel link, which leads proper design consideration to achieve maximum gain when APCP and PDR are applied. A longer block imposes lower overhead during

segment composition, including the LTP header, but it is more prone to corruption in high BER channel conditions. Given 2k segmentation size and high BER, e.g., 10^{-5} , 10^{-6} BER, the majority of LTP session closings take 5 OWLT rounds. In BERs lower than 10^{-7} , the LTP session closings are evenly distributed at 3 OWLTs and 5 OWLTs and have no meaningful improvement of shortened session delivery time with APCP and PDR. So, the extremes of the effective region of design, BER 10^{-5} and 10^{-6} , only add redundancy. Note that BER 10^{-5} and 10^{-6} are quite common in space missions.

During simulation, severely long latency cases of LTP session closing are observed even under very low BER, 10^{-8}

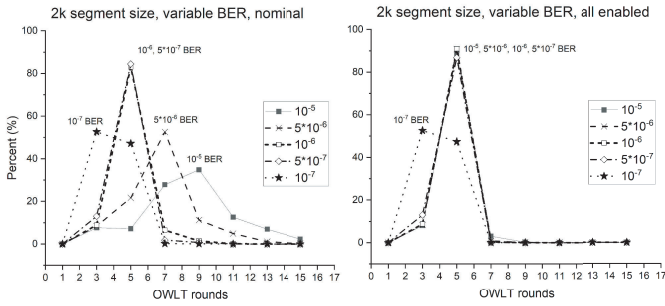


Fig. 8. Comparison of segment loss rates for 2k segment size, in nominal operations versus with all upgrades, under various BER.

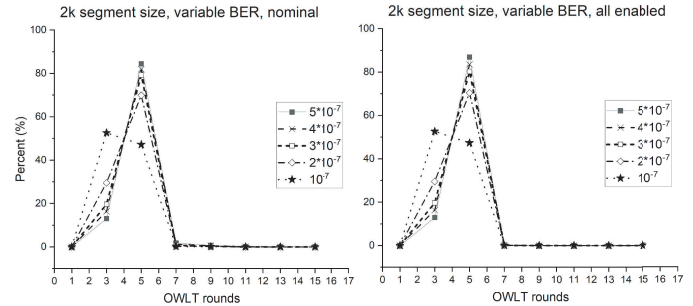


Fig. 10. Comparison of 2k segment size and nominal & all enabled between 5×10^{-7} and 10^{-7} .

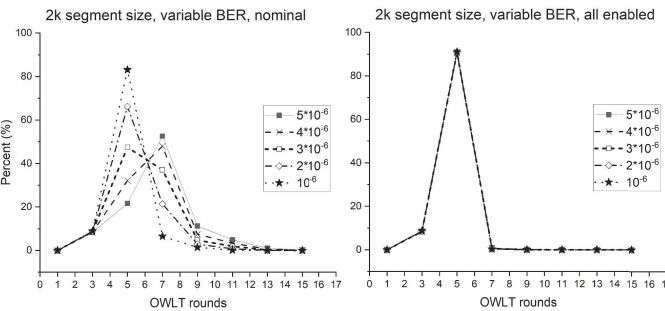


Fig. 9. Comparison of 2k segment size and nominal & all enabled between 5×10^{-6} and 10^{-6} .

and 10^{-9} .

There is no published analysis of the maximum LTP session closing latency that can be tolerated on mission-critical operations in deep space. Closing time longer than 9 OWLTs is not negligible because it makes ground or spacecraft waiting unexpected, which can lead to violation of real-time processing requirements or early time-out during segment reception (causing segments to be abandoned). In this situation, real-time negotiation between comm assets separated by long distances will be severely compromised. If APCP and PDR are used in all LTP sessions, the chances of extremely large session closure latency are reduced significantly. However, there are still chances it will happen.

Using PDR, goodput drops by 12.45% at 10^{-5} BER. However, it is decreased to less than 1.6% in equal or less than 10^{-6} BER, which is a marginally acceptable cost compared to the severity of extremely long session closure latency.

In conclusion, applying APCP and PDR mechanisms to 1k segment size & 10^{-5} BER or to 2k segment size & 10^{-5} and 10^{-6} BER yields meaningful improvements regarding the LTP session closing time. Also, APCP and PDR significantly reduce the possibility of requiring more than 9 OWLT rounds to close a session. This mechanism is equivalent to a 10x improvement on BER in 1k & 2k segment size from 10^{-5} to 10^{-6} BER with a 12.45% goodput drop and from 10^{-6} to 10^{-7} BER with a 1.6% goodput drop. We think it is more effective and practical than directly enhancing BER or decreasing the data rate.

VI. CONCLUSION

This study presented detailed simulation results for the temporal behavior of LTP session closing time as affected by segment loss. The aggressive and proactive LTP signal management scheme and provisional DS retransmission method for enhancing bounded session completion time were proposed, and the corresponding behavior has been assessed through temporal simulations. The simulation results show that in nominal LTP application and operations, LTP session closing time can reach intolerable ranges in non-negligible cases, which hinders a system from working in a timely manner in terms of real-time operation. The latency of the LTP session closing time is mainly determined by OWLT. Some techniques that can be leveraged to reduce an LTP session closing time in a real operation environment were suggested. The effects of the suggested methods were assessed via simulation results. It is observed that at 10^{-6} BER, 99% of LTP sessions are completed within 5 OWLTs with the schemes suggested in this study, whereas only 92% of LTP sessions in the original mechanism of nominal LTP are.

We have found that the progressive and proactive LTP control mechanism successfully limits the LTP session closing time to 3–5 OWLTs in most cases. Even though the probability of data corruption in the retransmitted DSs by the progressive and proactive LTP control mechanism repeatedly during retransmission is very low, that probability cannot be excluded. As a result, the LTP session closing time can reach 7, 9, or 11 OWLTs, which may be operationally intolerable. However, this has a very low probability of happening. This mechanism is particularly effective under conditions of high BER, e.g., 10^{-5} , and big segmentation size. Although it imposes an overhead of 6–10% in bandwidth throughput, we believe it is useful for securing stable mission operation in deep space where the tolerable range of data delivery time is more crucial than the total amount of data delivered regardless of delivery completion time.

The throughput penalty of this scheme under the same conditions is only 1.5%. Therefore, the suggested scheme should be of interest to deep space missions where light distance is comparatively longer than for low earth orbit (LEO) missions, but real-time operation function is nonetheless highly beneficial. Adaptive tailoring of this scheme to anticipated space environments is appropriate for effectively leveraging

this scheme to its operating environment to achieve desirable limitation of LTP session closing latency with acceptable throughput penalty because, in a very high-error channel, the throughput penalty can grow unacceptably. Our analysis shows this scheme is effective when BER is less than 6×10^{-6} . Possibly a strong erasure coding scheme might be more effective than this scheme in a very noisy channel; leveraging this scheme is advantageous in real-time operations in most applicable space communication environments.

Applying this scheme to real applications may show different results in environments in which hardware performance, queuing buffer handling mechanism and message delivery priority are different. Also, variations are expected depending on whether the segment data queuing buffer is highly congested or allows instantaneous service. Those variable factors shall be considered in a following study and future study will be targeted to prove the effectiveness of this scheme through an interoperability test with relevant applications.

REFERENCES

- [1] NASA, "DTN resources for mission development," [Online]. Available: https://www.nasa.gov/directorates/heo/scan/engineering/technology/disruption_tolerant_networking_resources
- [2] N. Alessi, S. Burleigh, C. Caini, and T. De Cola, "LTP robustness enhancements to cope with high losses on space channels," in *Proc. ASMS/SPSC*, Oct. 2016.
- [3] CCSDS, "Erasure correcting codes for use in near-Earth and deep-space communication," *CCSDS 131.5-O-1*, 2014.
- [4] H. Wu, Y. Li, J. Jiao, and Q. Zhang, "LTP asynchronous accelerated retransmission strategy for deep space communications," in *Proc. IEEE WiSEE*, 2016.
- [5] Q. Yu *et al.*, "Modeling RTT for DTN protocol over asymmetric cislunar space channels," *IEEE Syst. J.*, vol. 10, no. 2, pp. 556–567, Jun. 2016.
- [6] K. Zhao *et al.*, "Performance of bundle protocol for deep-space communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2347–2361, May 2016.
- [7] B. Cao *et al.*, "Expected file-delivery time of DTN protocol over asymmetric space internetwork channels," in *Proc. IEEE WiSEE*, 2018.
- [8] W. Zhang *et al.*, "Licklider transmission protocol for GEO-relayed space internetworking," *Wireless Netw.*, vol. 25, no. 7, pp. 3747–3757, Feb. 2018.
- [9] L. Yang *et al.*, "An analytical framework for disruption of Licklider transmission protocol in Mars communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5430–5444, Feb. 2022.
- [10] CCSDS, Next generation uplink, *CCSDS 230.2-G-1*, 2014.
- [11] 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.
- [12] Z. Shi *et al.*, "Study on checkpoint timer optimal setup of Licklider transmission protocol," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 35, no. 5, pp. 4–13, May 2020.
- [13] S. Burleigh, "Interplanetary overlay network: An implementation of the DTN bundle protocol," in *Proc. IEEE CCNC*, 2007.
- [14] R. M. Jones, "Disruption tolerant network technology flight validation report DINET," *JPL Publication 09-2*, 2009.
- [15] A. Jenkins, S. Kuzminsky, K. K. Gifford, R. L. Pitts, and K. Nichols, "Delay/disruption-tolerant networking: Flight test results from the international space station," in *Proc. IEEE AeroConf*, 2010.
- [16] K. Nichols *et al.*, "DTN implementation and utilization options on the international space station," in *Proc. SpaceOps*, 2010.
- [17] A. Schlesinger, B. M. Willman, L. Pitts, S. R. Davidson, and W. A. Pohlchuck, "Delay/disruption tolerant networking for the international space station (ISS)," in *Proc. IEEE AeroConf*, 2017.
- [18] NASA, "Disruption tolerant networking experiments with optical communications," [Online]. Available: https://www.nasa.gov/directorates/heo/scan/news_DTN_Experiments_with_Optical_Communications.html
- [19] D. Israel *et al.*, "The benefits of delay disruption tolerant networking (DTN) for future NASA science missions," in *Proc. IAC*, 2019.
- [20] J. Esper, "Draft LunaNet interoperability specification," V003, [Online]. Available: <https://esc.gsfc.nasa.gov/staticfiles/Draft%20LunaNet%20Interoperability%20Specification%20Final.pdf>
- [21] W. Tai, "IOAG Lunar communications architecture study," [Online]. Available: <https://trs.jpl.nasa.gov/bitstream/handle/2014/50116/CL%202318-3136.pdf?sequence=1&isAllowed=y>
- [22] IPNSIG, [Online]. Available: <https://ipnsig.org>
- [23] R. C. Durst, P. D. Feighery, and K. L. Scott, "Why not use the standard Internet suite for the interplanetary Internet?," [Online]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.8972&rep=rep1&type=pdf>
- [24] V. Kushwaha and R. Gupta, "Delay tolerant networks: Architecture, routing, congestion, and security issues," in *Handbook of research on cloud computing and big data applications in IoT, IGI Global*, pp. 448–480, 2019.
- [25] R. Wang, T. Wang, and X. Wu, "Bundle protocol (BP) over Licklider transmission protocol (LTP) for cislunar communications," in *Proc. IEEE GLOBECOM*, 2009.
- [26] R. Wang, S. C. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 359–368, Apr. 2011.
- [27] R. Zhang *et al.*, "Licklider transmission protocol for GEO-Relayed space networks," in *Proc. MLCOM*, 2017.
- [28] N. Bezirgiannidis and V. Tsaoussidis, "Packet size and DTN transport service; Evaluation on a DTN testbed," in *Proc. IEEE ICUMT*, 2010.
- [29] R. Wang *et al.*, "The effect of "window size" on throughput performance of DTN in lossy cislunar communications," in *Proc. IEEE ICC*, 2012.
- [30] 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.
- [31] M. Ramadas, S. C. Burleigh, and S. Farrell, "RFC 5326, Licklider transmission protocol specification," IRTF DTN Research Group, [Online]. Available: <https://tools.ietf.org/html/rfc5326>
- [32] 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, 2007.
- [33] NASA, "Interplanetary overlay network (ION) design and operation," ver 3.7.2, JPL D-48259, [Online]. Available: <https://sourceforge.net/projects/ion-dtn/>
- [34] C. H. Koo and S. C. Burleigh, "Structural considerations for generating and handling LTP report segments from an interoperability testing," *J-KICS*, vol. 47, no. 12, pp. 2065–2077, Oct. 2022.
- [35] N. Bezirgiannidis, S. Burleigh, and V. Tsaoussidis, "Delivery time estimation for space bundles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 3, pp. 1897–1910, Jan. 2013.
- [36] R. Lent, "Analysis of the block delivery time of the Licklider transmission protocol," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 518–526, Jan. 2019.
- [37] Wikipedia, "Bit error rate," [Online]. Available: https://en.wikipedia.org/wiki/Bit_error_rate



Cheol Hea Koo was born in Boeun, Chungbuk, Republic of Korea in 1971. He received the B.E. and M.E. degrees in electronic engineering from the Chungnam National University in 1997 and 1999 respectively, and the Ph.D. degree in Computer Engineering from Chungnam National University in 2021. Currently, he is a Principal Researcher at Korea Aerospace Research Institute. His research interests include real-time embedded software, space communication protocol development, delay-tolerant networking technology implementation.



Scott C. Burleigh He also coauthored the specification for version 6 of the DTN Bundle Protocol (BP, Internet RFC 5050), supporting automated data forwarding through a network of intermittently connected nodes, and was Lead Author for the specification for BP version 7, RFC 9171. Mr. Burleigh led the development of implementations of BP, LTP, and related protocols that are designed for integration into deep space mission flight software and are currently in operation on the International Space Station, with the long-term goal of enabling deployment of a delay tolerant Solar System Internet. Mr. Burleigh has received the NASA Exceptional Engineering Achievement Medal and four NASA Space Act Board Awards for his work on the design and implementation of these communication protocols.