Published in final edited form as:

IEEE Trans Commun. 2019 January ; 67(1): 518–526. doi:10.1109/TCOMM.2018.2875717.

Analysis of the Block Delivery Time of the Licklider Transmission Protocol

Ricardo Lent [Senior Member, IEEE]

University of Houston, Houston, TX 77004 USA.

Abstract

The Licklider transmission protocol is a point-to-point communication protocol designed for space links, which commonly involve extreme delays, disruptions, and lossy transmissions. The protocol sends application data in blocks, which in turn are sent in segments. It achieves reliable block delivery through multiple transmission rounds, each one re-sending the segments lost during the previous round. This retransmission process drives protocol performance. We derive exact and approximate methods to find the average number of rounds per block. Then, we estimate the block delivery time and other metrics using this value. We found that the common practice of matching segment lengths to the maximum transfer unit of the link layer may lead to suboptimal performance. The models provide accurate protocol performance prediction, which can help to optimize protocol parameters for specified operating conditions.

Index Terms

Communication system performance; data communication; networks; reliability; space vehicle communication

I. Introduction

EXTREME propagation delays, high bit error rates, asymmetrical bandwidth, power limitations, and periodic channel disruptions are some of the main issues that make space communications challenging. Among other factors that contribute to the problem are the limited capacity of the spacecraft, occultation, and the need for autonomy. For example, a communication between Mars and Earth involves a delay of 3–24 minutes depending on the location of the planets. In contrast, a transmission from Pluto requires about 5 hours one-way at light speed. The intensive use of acknowledgments (ACK) in communication protocols, which account for most protocols designed for terrestrial use, suffer from very poor performance under such conditions. These reasons have motivated the development of a Delay-Tolerant Networking (DTN) approach to Interplanetary Internet [1].

The Licklider Transmission Protocol (LTP) is part of the DTN protocol stack that was developed to provide point-to-point delay-tolerant data transmission under challenging conditions. LTP transmits *blocks* of application data using a number of segments. Portions of

a block may be required to be transmitted reliably whereas some other portions not. The protocol defines procedures for the receiver to request the sender the retransmission of lost and discarded segments. This feature allows LTP to support both best effort and reliable modes [1]. LTP has been implemented as part of the *Interplanetary Overlay Network* (ION) [2] and has been flight tested in recent years [3], [4].

LTP performance depends on multiple factors that can be controlled to some degree by the network operator. For instance, the choices of transmission power levels, link layer protocol, forward error correction codes, and modulation types, are relatively constrained, but allow some control of the physical channel to achieve a suitable bit error rate (BER). However, the BER alone is only one factor that conditions LTP performance. Within LTP itself, it is also possible to set multiple parameters that have an impact on performance, such as the data segment length, timer deadlines, and block sizes. Tuning these parameters require either (or both) high operators' expertise and accurate LTP modeling.

To attend to this need, this work contributes with the:

1. Analysis of the average number of rounds required for the reliable transmission of a LTP block. Using a first-principles approach, we derive both exact and approximate formulae for this metric. The approximation gives better accuracy than comparable methods that have appeared in the literature.
2. Estimation of the block delivery time and block service time, considering the contribution of the red-part of the block size, the use of a forward error correction code, and the choice of buffer size and aggregation time limit. Limited experimental results have been reported to date on some of these aspects.
3. Evaluation of the performance impact of segments that span more than one link layer frame. No previous works have addressed this possible configuration choice.
4. Analysis of the optimal segment payload length that minimizes the average block delivery time under a specified choice of LTP parameters and channel features, including the BER, propagation delay, and transmission rate.

The rest of the paper is organized in the following way. Next section discusses the current state-of-the-art in performance evaluation of LTP. In section III, we provide a brief overview of LTP. We develop LTP performance models in section IV and contrast model predictions with simulation results in section V. In section VI, we discuss the selection of the optimal segment length. Finally, section VII provides closing remarks on this work.

II. Related works

Space communications involve many challenges that are associated with extreme propagation delays, high bit error rates, asymmetrical bandwidth, and long channel disruptions [5] that limit the correct operation of most terrestrial communication protocols. New protocols and enhancements to existing ones have appeared in recent years to address these challenges. The performance evaluation of the new protocols has been of high interest [6]. Several works have addressed the use of Internet Protocols for space communications,

for example, for satellite-based communications [7]. Caini *et al.* [8] studied the use of TCP supporting delay-tolerant networking. They concluded that the use of a bundle layer improves retransmission performance when channel disruptions can occur. Similar conclusions were drawn by Wang *et al.* [9] from experiments conducted with LTP over TCP and UDP, as compared to the sole use of TCP [9] both over single hops and end-to-end paths [10].

The closest works in the literature to our analysis were conducted by Yu *et al.* [11], Yang *et al.* [12], and Wu *et al.* [13]. In these works, the authors analyzed the performance of LTP over lossy channels, identifying the main elements that contribute to block transmission. While related to ours, prior works were based on substantially different modeling methods to determine the transmission effort. In the latter work, the authors also proposed an accelerated retransmission strategy to improve delivery times. Another close work was done by Lu *et al.* [14], who examined the optimal packet length of a DTN transmission over multiple hops. We develop more accurate methods to find the average number of block rounds than the ones used by these works. One more related work was conducted by Bezirgiannidis and Tsoussidis [15] who experimentally determined the benefits of using large LTP segments—as close as possible to the length to the maximum transmission unit of the link layer. In another work, Bezirgiannidis *et al.* [16] also analyzed the delivery time of bundles over a path.

The impact of bundle aggregation into LTP blocks is another important performance factor that has received some attention. It was experimentally studied by Wei *et al.* [17], whose results suggested the benefits of aggregating multiple bundles into larger blocks when the channel bit error rates are low. The study also indicated that no observable benefits resulted from a multi-bundle aggregation with high BER. However, the results also suggest that the study did not consider session concurrency. A separated work addressed the evaluation of concurrent block sessions [18]. The study was only experimental, but concluded that the number of bytes per session (NBS or block size), does not impact performance severely, but the number of sessions (NOS) does. The authors suggested that better throughput could be achieved with smaller blocks to have a larger NOS for a given transmission. In another study, the memory usage aspects of LTP were evaluated [19].

The public availability of the *Interplanetary Overlay Network* (ION) software implementation has fostered various experimental studies. Particular performance aspects, such as the impact of link delays [20] and goodput [21] have been investigated. Also, extensions to improve LTP performance have been reported, including LTP-T [22], the use of proactive retransmissions of signaling segments and use of a list of recently closed sessions [23], and the integration of Reed-Solomon codes to LTP [24]. Collectively these works provide the current context for our analysis.

III. Overview of the Licklider Transmission Protocol

What follows is a basic description of the operational aspects of the Licklider Transmission Protocol (LTP). Further details can be found in the related RFCs [25], [26]. LTP operates as an intermediate protocol in the DTN stack. It receives data from an upper layer protocol for

transmission. Typically, LTP operates below the Bundle Protocol (BP), so it normally receives *service data units* from BP. LTP stores this upper layer data in blocks and operates by sending these blocks to the receiving end through a lower layer protocol, commonly, a link-layer protocol. Let us call *frames* the link-layer packets. There is no restriction on what a block can transmit, so depending on the application data, it could contain a whole bundle, part of a bundle, or even multiple bundles depending on the amount of memory reserved for a block, i.e., the block size. A portion of the block is labeled *green-part* and the rest *red-part* to indicate a best effort of reliable delivery mode respectively.

The transmission of a block occurs as a sequence of fragment transmissions called data segments (DS). The process is called a *session*. The size of a segment is commonly fixed to match the value of the maximum transmission unit (MTU) parameter of lower-layer protocol(s). However, there is no particular limitation on the size of segments, so that they could use less than the MTU, or it could even span multiple low-layer frames. As we will see later, we analyze both cases to highlight the performance benefits of adjusting segment lengths depending on the operational conditions.

A session—the communication of a single block transmission, involves at least one transmission round. During that round all of the block's segments are radiated. The last segment is labeled the *checkpoint segment* (CS) that carries the flag *end-of-red-part* (EORP). This flag signals the receiver that it needs to respond with a report segment (RS). The RS acknowledges reception of the DSs and indicates the sender which segments require retransmission, so it functions both as a positive and negative acknowledgment mechanism for the whole block. Additional rounds then proceed until all of the segments are either successfully received or the maximum number of rounds is reached. In the second case, the block is discarded. On reception of the final confirmation (RS) from the receiver, the sender transmits a report acknowledgment (RA) to close the block transmission. LTP blocks may be transmitted one at a time, i.e., transmission of a block starts only after the previous completely finishes, or in multiple concurrent sessions. The latter mode offers better performance but may be limited by the memory availability to LTP at both ends.

IV. LTP Block Transfer Analysis

The protocol using LTP, such as the Bundle Protocol, may (periodically) ask the LTP engine to transmit data. As data arrives to the LTP engine for transmission, data is first accumulated in a *block*. LTP operates by transmitting data blocks to the remote LTP instance. We analyze the factors that impact the delivery time of a LTP block. To make the discussion easier to follow, Table I summarizes the main parameters that we use in the analysis, clustered by their approximate relation to the protocol stack.

A. Deferred Start Time and Average Block Size

The start time of a block transmission may be deferred under two conditions. The LTP engine must wait for the channel to become accessible, which may add delay if the channel is not immediately ready. In space communications, occultation prevents the line-of-sight between the communication pair, making the channel temporarily unavailable. Also, considering that the communication equipment constitutes an expensive resource in the

spacecraft, it is normally used for multiple purposes. The channel then may become intentionally unavailable to the LTP engine to serve a different purpose, which also adds delay to the block transmission. Let us call w_1 the average waiting time ($w_1 \geq 0$) for the channel. The calculation of this value depends on various mission-specific factors, such as the ephemeris and the trajectories of the sender and receiver.

The LTP engine may select to defer on purpose the start time of a block transmission until either a time limit (b^o) or an aggregation data limit (b^+) is met. Let the average block size be denoted by b , $0 < b \leq b^+$. Buffering parameters b^o and b^+ are user (operator) configurable. Let w_2 be the average delay ($w_2 \geq 0$) introduced by buffering data at the LTP engine. To find w_2 , we need to consider the intensity at which data arrives at LTP for transmission. If λ represents the application load (in bps), the deferral time w_2 is given by:

$$w_2 = \begin{cases} b^+/\lambda; & \lambda \geq (b^+/b^o) \\ b^o; & \text{otherwise} \end{cases}$$

(1)

Similarly, we can find the average block size b , which also depends on the specific LTP engine configuration and the application load:

$$b = \begin{cases} b^+; & \lambda \geq (b^+/b^o) \\ b^o\lambda; & \text{otherwise} \end{cases}$$

(2)

The union of w_1 and w_2 gives $w = w_1 + w_2$ the first source of delay for a block transfer.

B. Block Delivery Time

It is the time needed to deliver all bits of the block to the receiver, including any retransmission of red-part segments. A LTP session sends a block in one or more rounds. In each round the LTP sender sends a sequence of data segments (DS) to the LTP receiver. All of the segments comprising the block are sent during the first round. In the nominal case (no segment losses or rejections), the block delivery time is just the radiation time of the segments plus the propagation delay $d \geq 0$. From the LTP perspective, there is no distinction between a segment error, loss, or rejection, so we will use these terms interchangeably. Additional rounds may be needed to handle the retransmission of lost segments from the red-part of the block. These retransmissions are required by the receiver through the report segment (RS), which is sent at the end of each round. Also, a local timer ensures the retransmission of CP segments, RS, and acknowledgments in case any of them is lost. Fig. 1 depicts typical transmission scenarios. A cross ("X") indicates segment loss. Block delivered

times are indicated by the discontinuous (blue) vertical lines. The exact dynamics of case E depends on the implementation and the value of the timeout interval T_o at each end. The discontinuous lines indicating RS and RA transmissions show a second possible behavior. However, either choice produces about the same effect to the number of rounds and block delay. The average block delivery time (T) consists of three components in addition to the deferring time w :

$$T = w + d + T_r + T_p$$

(3)

where d is the propagation delay between the sender and receiver, T_r is the total segment radiation (transmission) time, and T_p is the time penalty introduced by the extra rounds that handle the retransmission of rejected segments (see sections IV-D and IV-E). Delay d is the transmitter-receiver distance D divided by the radio propagation speed—usually the speed of light c ($d = D/c$). We assume that the model is intended for the study of cases with long propagation delays, so that we can neglect the contribution of processing delays and the radiation time of small control segments.

Also, note that we have excluded the case of a RS or RA loss after all DS have been successfully delivery. These cases correspond to those labeled E and F in Fig. 1. A RS loss involves a timeout and the retransmission of the previous CP segment with an extra RS/RA round. A RA loss also adds a timeout and additional RS/RA round. Neither case affects the block delivery time.

C. Block Service Time

A related metric is the block service time, which is the time needed by the sender to handle a block session. Unlike the block delivery time, the service time S is affected by the loss of the last RS because the sender requires the report to close the session—see case E in Fig. 1, case F has negligible impact as before. Extending (3) and ignoring the small contribution of the RS transmission time, the average service time (S) is:

$$S = T + \frac{p_r}{1 - p_r} T_o$$

D. Total Segment Radiation Time (T_r)

The total segment radiation time T_r is the time spent within the LTP session transmitting (from both the red-part and green-part) and retransmitting segments (from the red-part only). Let m be the selected amount of payload, in bits, that can be carried by one data segment. Assuming the usual condition, $b \gg m$, so that all data segments (with the possible exception of the last one) are of the same size. The average number of segments that require transmission is $n = b/m$, where b is the average block size as before. To determine the number of segment retransmissions, consider p to be the segment loss probability and f the

average fraction of the block that is red-part. The value of f depends on the application and ranges from 0 (no red-part) to 1 (no green-part). The total average segment radiation time is:

$$T_r = \frac{L}{r} \frac{b}{m} \left[(1-f) + \frac{f}{1-p} \right]$$

(4)

where r is the channel bit transmission rate and L is the frame length(s). Within this context, “frame” refers to the packet that is physically transmitted. The term $nf/(1-p)$ is the average number of transmissions needed to deliver nf red-part segments under the assumption of uncorrelated segment loss probability p . To be precise, T_r should be extended with an additional term to account for the radiation time of the RS, which usually occurs at a much smaller rate on the return channel. However, compared to the rest of terms, the contribution of the RS radiation times is minimal, so we have exclude them from (4) for the sake of simplicity.

A more significant aspect to evaluate is the possibility of mapping a LTP segment to more than one lower layer frame. We extend the analysis to determine L and p in this case.

1) Probability of Frame Error: Let us examine the probability of frame error under the assumption of uncorrelated channel bit errors, which occur with probability p_e , i.e., the bit error rate (BER). A frame of length L is lost with probability p_f :

$$\begin{aligned} p_f &= P\{\text{frame error}\} \\ &= P\{\chi \text{ bit errors in the frame}\} \\ &= 1 - \sum_{i=0}^{\chi} \binom{L}{i} (1-p_e)^{L-i} p_e^i \end{aligned}$$

(5)

where χ is specific to the forward error correction (FEC) code used by the lower-layer protocol(s). Stronger FEC codes allow the successful decoding of frames with a larger number of bit error, but they also add longer parity sequences to the frame, so that the frame header h_f becomes larger. If no FEC is used, then a single bit error is sufficient to invalidate the frame, in which case:

$$\begin{aligned}
 p_f &= P\{1 \text{ bit error in the frame}\} \\
 &= 1 - (1 - p_e)^L = 1 - e^{L \cdot \log(1 - p_e)}
 \end{aligned}$$

(6)

Note that under uncorrelated bit errors, the probability of frame loss is an independent event. To illustrate the point consider the case of no FEC. The probability of event x : “successful delivery of segment 1” given event y : “successful delivery of segment 2” is: $P(x/y) = (1 - p_e)^{2L/(1-p_e)^L} = (1-p_e)^L = P(x)$, that is, the probability of x does not depend on y .

2) Probability of Segment Loss: With respect to p (probability of segment loss) and L (frame length), we have two possible cases depending on the LTP engine configuration:

1. If one segment maps to one frame, then the segment payload m is limited by the lower-layer MTU m^+ minus the space needed for the LTP header h , which leads to:

$$L = m + h + h_f; \quad 0 < (m + h) \leq m^+$$

(7)

where h_f is the lower-layer header(s). The segment loss probability is just the frame loss probability: $p = p_f$ for a frame of length L .

2. If one segment maps to multiple frames, then m is only limited by the amount of data to be sent, i.e., b . The number of frames per segment is at least $(m + h)/m^+$, but it could be more given that the maximum segment size could be less than the MTU. Assuming the best case:

$$L = m^+ + h_f; \quad 0 < m^+ < (m + h)$$

(8)

The segment loss probability depends on the probability of losing one or more of the constituting frames of the segment.

3) Multi-frame Segments: At this point, it is interesting to ask the question of whether it makes any sense to transmit a segment using more than one frame. Sending the block using multiple segments involves sending separate headers per segment, which creates

protocol overhead. A clear benefit of multi-frame segment transmissions is to save on LTP headers. If reliable transmissions are required, losing one or more frames in the former case involves the retransmission of just the missing frames as opposed to all of the frames as required in the latter case as LTP does not provide any special error recovery mechanism for those cases.

To illustrate the point, consider two opposite cases in the reliable transmission of a block of size b , with $b \gg m^+ \gg h, h_f$ and no FEC:

- Case I: the block is sent in segments of length $(m^+ - h)$, i.e., the size of the MTU minus LTP's header length. For reliable transmission, the average number of segment to be sent is $b/[(m^+ - h)(1 - p_A)]$, where $p_A = p_f(m^+ + h_f)$ —that is, (6) evaluated with $L = m^+ + h_f$. The total amount of data sent in this case is:

$$\frac{b}{m^+ - h} \frac{1}{1 - p_A} (m^+ + h_f)$$

(9)

- Case II: the block is sent as a single segment of length b . For reliable transmission, the average number of segments sent is $(b + h)/[m^+(1 - p_b)]$, where $p_b = p_f(b + h)(1 + h_f/m^+)$. The total amount of data sent in the second case is:

$$\frac{b + h}{m^+} \frac{1}{1 - p_b} (m^+ + h_f)$$

(10)

The use of multi-frame segments is more efficient than the use of single-frame segments if and only if the value of expression (10) is less than expression (9). After some algebraic manipulation and considering that $b/(b + h) \approx m^+/(m^+ - h) \approx 1$, one can conclude that such case is only beneficial under the condition $P_A > P_b$. It is straightforward to see that this expression does not hold given that P_b is larger than P_A as $b \gg m^+$, so the use of multi-frame segments is not recommended.

E. Time Penalty for Retransmission Rounds (T_p)

The fourth and last component of the block delivery time is the time introduced by the retransmission rounds. This time penalty applies only to the transmission of the red-part of the block, i.e., to the transmission of:

$$n = f \frac{b}{m}$$

(11)

segments. Let κ be the average number of rounds needed to transmit n segments. Clearly at least one round is needed, so $\kappa \geq 1$. Additional rounds happen because of the following two possibilities:

1. A round may start upon reception of a report segment. In this case, the round adds $2d$ (two times the propagation delay) to the block delay. If p_c and p_r are respectively the frame loss probabilities of the CP segment and RS, then this event occurs with probability $(1 - p_c)(1 - p_r)$, since it can only happen when both segments arrive error free (see Fig. 1-B).
2. A round may be triggered after rejecting a CP segment or RS (see Fig. 1-C,D). In either case, after the timer for those segments expire, it adds T_o to the block delivery time, observing that $T_o > 2d$ must hold to prevent early retransmissions:

$$T_o = 2d + \xi$$

(12)

where ξ is a constant and includes at least RS radiation time. A timeout event happens with probability:

$$p' = 1 - (1 - p_c)(1 - p_r).$$

(13)

The values of p_c and p_r can be obtained with expression (5) (or (6) without a FEC). Depending on the value of b , the first CP segment may be smaller than other DS. The exact payload is $(fb - \lfloor n \rfloor m)$. However, subsequent rounds can involve CP traveling with maximum DS lengths. We can associate p_c to p : $p_c = ep$. A reasonable value for e is 1. Lower values, $0 < e < 1$ could improve accuracy for particular cases, so e can be used as a model tuning parameter. Note that in Fig. 1 (D), the second DS “n” carrying the CP is ignored by the receiver because of the RS transmission. The time penalty introduced by the extra rounds is then:

$$T_p = (\kappa - 1)[(1 - p')2d + p'T_o] = (\kappa - 1)(2d + p'\xi)$$

(14)

We propose both exact and approximate methods to calculate the average number of session rounds κ .

1) Average Number of Session Rounds: Exact Method: To achieve the reliable transmission required by the red-part of a block, LTP persistently retransmits undelivered segments through a series of transmission rounds. During the first round, all n red-part segments are sent and each subsequent round retransmits the set of unsuccessful segments of the previous round. The process stops once all segments have been successfully delivered. Let X_k be the process state defined as the number of segments to be transmitted during round (or step) k . Conceptually, assume a last round with zero segments to transmit.

It is easy to see that by construction, the sequence of random variables $\{X_0, X_1, \dots\}$ has the Markov property because the conditional probability of the next state solely depends on the current state and not on how it was reached, that is, the steady-state probability q_{ij} of the transition from state i to state j is: $q_{ij} = P(X_{k+1} = j | X_k = i)$. As a result, $\{X_k\}$ is an embedded, absorbing Markov chain with epochs (time of observations) occurring at the beginning of each transmission round, state space $\{0, 1, \dots, n\}$ ($n + 1$ states), and single absorbing state 0 ($q_{00} = 1$). LTP block transmission process can be therefore described by this chain starting at state n , i.e., $X_0 = n$ (see Fig. 2), and after a certain number of k transitions, reaching $X_k = 0$ with probability 1. We assume that the process cannot be interrupted earlier by some other mechanism.

The chain is time-homogeneous, $P(X_{k+1} = j | X_k = i) = P(X_k = j | X_{k-1} = i)$, and the transition probability from state i to state j , $i \rightarrow j$ is given by:

$$q_{ij} = \begin{cases} \binom{i}{j} (1-p)^{i-j} p^j; & i \geq j \\ 0; & i < j \end{cases}$$

(15)

where p is the segment loss probability given by expression (5) or (6). The transition matrix for this chain is:

$$Q = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ q_{10} & q_{11} & 0 & \dots & 0 \\ q_{20} & q_{21} & q_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ q_{n0} & q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix}$$

The average number of steps for reaching the absorbing state 0 can be easily obtained using the fundamental matrix $U = (I - A)^{-1}$, where A is a $n \times n$ matrix of transition probabilities of the transient states $Q = q_{ij}$ for $i, j = 1, \dots, n$, and I is the identity matrix.

The exact number of steps to absorption starting from any state is $S = Uv$, where v is a unit column vector. The interest is in the average number of steps starting from n , which corresponds to the n -th element of vector S : S_n . The average number of rounds is affected by CS and RS losses given that the state can only move after both segments arrive. The average number of rounds is therefore:

$$\kappa = \frac{S_n}{1 - p'}$$

(16)

The downside of this method is that is computationally expensive, making it suitable for small values of n . The main limitation is in the digital representation of binomial coefficients, which can grow very large, easily exceeding the storage capacity of the *double* data type. However, it is possible to handle large coefficients using special techniques, such as the ones provided through *evalin* in MATLAB. The calculation for a block of n segments involves a space complexity of $O(n^2)$ with a time complexity of $O(n^{2.373})$ using optimized matrix inversion methods.

2) Average Number of Session Rounds: Approximate Method: It is possible to find an approximate solution using the average number of visits $\bar{v}_j$ to each transient state j in the chain. The average number of steps to absorption is the sum of $\bar{v}_j$ for all j . To estimate $\bar{v}_j$, consider a reordering of the transient states, so that the start state n appears in position 1, state 1 in position 2, and so on, and that p is small (e.g., $p < 0.1$). The first new state (representing n segment transmissions) would have the largest number of visits of all the transient states being the starting state. Since p is small, the probability of returning to the initial state can be ignored and the average number of visits for $j = 1$ would be $\bar{v}_1 = 1$. For the rest of states, it is sensible to expect an approximated geometric distribution in the number of visits, as considered in the Wright-Fischer model [27]. We have found that a simple expression that follows this observation is:

$$\bar{v}_j \approx j^{\log(p)}$$

(17)

where p is the probability of segment loss. Using (17), the mean time to absorption can be estimated by: $\sum_{j=1}^n \bar{v}_j = -\log_p(n) + \gamma$, where $\gamma = 0.5772 \dots$ is Euler-Mascheroni constant that results from the partial summation of reciprocals. Note that $\log(p)$ is always negative or zero. Therefore, the average number of rounds is approximately:

$$\kappa \approx \max \left\{ 1, \frac{-\log_p(n) + \gamma}{1 - p'} \right\}$$

(18)

3) Validation of Approximation Assumption and Exact Method: To validate the method, we implemented the basic LTP's transmission process as a computational model. The simulator keeps track of the number of segments delivered by each round starting from n segments and using an independent segment loss probability p . To keep the simulator simple enough to minimize the possibility of implementation errors, this simulator does not consider LTP control mechanisms, only the delivery, and retransmission behavior. To achieve high statistical accuracy in the estimation of the metrics of interest, we recorded a large number of simulation samples (20,000), for different values for n and p .

Fig. 3 (a) compares the average number of visits to state k obtained by the simulator and expression (18) with $p' = 0$. Note that the chart's ordinate is logarithmic. Contrasting the results, we find that (18) provides accurate estimates for small values of p , but results deviate significantly for larger values (above $p = 0.1$). We later show, using a high fidelity simulation of BP and LTP, that the impact of the model deviation for p is not high. It is also interesting to consider that space link budgets (and coding gain) aim to achieve low BER, so the model remains applicable in practice. Fig. 3 (b) depicts the simulated and exact average number of steps to absorption as given by the model, which corresponds to the average number of rounds with $p' = 0$, for different LTP sessions of up to 100 segments each and for different values of p . We can observe an exact match between the simulation results and the model predictions, which provide evidence of the validity of the embedded Markov chain approach.

V. Test Scenario

To assess the approximate model, we compared the model outputs to event-driven simulation results of BP/LTP. We used a Python port of an event-driven simulator [28] that we extended to implement the protocols' specifications, including LTP's finite state machines of the sender and receiver [26]. Simulation verification was achieved both through static and

dynamic testing. The former consisted in examining the structural properties of the code and in running software tests, both of which conducted as part of the software development process. We also applied dynamic testing to examine inputs-outputs at runtime and execution traces to further verify that the simulated behaviors correspond to the specifications.

For the test, we assume one contact time period between Earth and a device located in the vicinity of a selected object. The communication occurs through a lossy 128 Kbps channel, without coding gain, and under uncorrelated errors, so losses are decided packet-by-packet according to (6). The propagation delays used in the study are: 1.3 ms, 1.28 s, 818 s, 2,470 s, and 4,667 s, which are related to links between Earth and the International Space Station (ISS), Moon, Mars, Jupiter, and Saturn. Statistics of the transmission of at least 500 single-block bundles of 10^4 bytes each were collected to characterize block delivery times. Other parameters were fixed as follows: BP header length: 100 bytes, segment header length: 20 bytes, RA length: 7 bytes, $h_f = 0$, $\xi = 2$. Fig. 4 depicts the results obtained with the approximate model and simulation using segment lengths of (a) 500 bytes and (b) 1000 bytes. The average number of rounds (κ) obtained with the approximate model and with $e = \frac{2}{3}$ to find P_c yielded comparable results to the simulation—Fig. 5 (a), with a root-mean-square-error (RMSE) of 0.2920 overall. In contrast, methods from the literature produced overall RMSE values of 1.20 with *method 1* [11], [12] and 0.94 with *method 2* [14]—see Fig. 5 (b). Fig. 4 provides objective evidence of the validity of the approximated model for the complete intended domain.

VI. Optimal Segment Length

Given the potentially large propagation delays involved in space communications, an optimization of transfer times is of crucial interest. To summarize, the average block delivery time T is given by expression (3). The term T_r can be found with (4), which gives the total segment radiation time. The segment loss probability is given in general by (5) (or (6) if no FEC) under the usual case where a segment fits within one frame. The frame length L is given by (7). Term T_p is given by (14) jointly with expressions (11), (12) and (13). Finally, the value of κ is approximated with (18).

The choice of data segment length may play a significant role in the value of T as it is one of the factors that determine the segment loss probability and consequently the average number of rounds needed to transmit a block in addition to the header overhead (ratio of total DS header transmissions to block size). The latter is $\frac{h+h_f}{m} \left[(1-f) + \frac{f}{1-p} \right]$. Larger m decreases the left-most term, reducing the overhead, but also increases p and the right-most term. To illustrate a concrete case, consider the reliable block transmission over the hypothetical Mars–Earth link of the previous section. We observe that shorter segments than the MTU may lead to lower block transfer times under given conditions as depicted in Fig. 6 (a). Assuming that the default MTU is 1500 bytes, the model predicts that it is possible to cut in more than half the transfer time of blocks when the channel BER is 10^{-5} or higher. Under low channel BER, larger segments produced the best results given the savings obtained by the lower header overhead. There is a clear trade-off between header overhead and the

number of retransmission rounds to consider when optimizing the segment length. Contrary to the common practice of assuming the default lower layer's MTU as the segment size, such value should be considered based on the current channel conditions.

More formally, we are interested in solving the constrained optimization of LTP payload's length to minimize the block delivery time, that is, we need to determine m^* :

$$m^* = \underset{m}{\operatorname{argmin}} T, \quad s.t. \quad 0 < m \leq m^+.$$

(19)

While it is difficult to find a closed-form expression for (19), the value can be easily calculated numerically using (3). Fig. 6 (b) depicts the optimal segment length for the 100 MB block.

In general, the optimal segment length mainly depends on the block size and channel features: the radiation rate, propagation delay, and bit error rate. We can observe that with small propagation delays it is in general better to use larger segments given that the aggregated penalty of the extra rounds is generally not sufficiently high to counteract the aggregated impact of the header overhead. The situation tend to be the opposite with longer links. In the cases of Mars, Jupiter or Saturn, smaller segments help to reduce the probability of segment error, which benefits the transmission delay by limiting the number of extra transmission rounds.

VII. Conclusion

We have analyzed LTP and have proposed both exact and approximate methods to calculate the average number of rounds for reliable block transfer under a specified set of LTP parameters, channel characteristics, and operating conditions. Using the model, we found the average delivery and service time for a block. We also determined the optimal segment length that achieves the lowest transfer times and observed that the common practice of assigning the maximum transfer unit of the link layer as the segment length often leads to suboptimal performance in lossy channels. Likewise, we demonstrated that the use of large segments that span multiple frames also leads to suboptimal performance. The proposed methods yield greater prediction accuracy than prior works and provide the basic information needed to derive other performance metrics with precision.

Acknowledgment

The author would like to thank G. Clark and S. Burleigh for their advice on the technical issues examined in this paper.

This work was supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program. The associate editor coordinating the review of this paper and approving it for publication was J. Luo.

Biography

Ricardo Lent (M'92–SM'14) received the B.Sc. degree in electronic engineering from Ricardo Palma University, Peru, the M.Sc. degree in telecommunications engineering from the Universidad Nacional de Ingeniería, Peru, and the M.Sc. and Ph.D. degrees in computer science from the University of Central Florida, USA, in 2003. He was a Research Fellow with the Department of Electrical and Electronic Engineering, Imperial College London, U.K., a Principal Network Engineer for industry and a Visiting Assistant Professor with the University of Central Florida. He is currently an Assistant Professor with the Department of Engineering Technology, University of Houston. His current research interests include cognitive networking, mobile and pervasive communications, and green computing. <http://www.uh.edu/~rlent/contact.html>

References

- [1]. Burleigh S et al., "Delay-tolerant networking: An approach to inter-planetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, 6 2003.
- [2]. The Interplanetary Overlay Network (ION) Software Distribution. IONDTN. Accessed: Apr. 1, 2018 [Online]. Available: <https://sourceforge.net/projects/ion-dtn>
- [3]. Jenkins A, Kuzminsky S, Gifford KK, Pitts RL, and Nichols K, "Delay/disruption-tolerant networking: Flight test results from the international space station," in *Proc. IEEE Aerosp. Conf.*, 3 2010, pp. 1–8.
- [4]. Ackerman E, "Interplanetary Internet tested," *IEEE Spectr*, vol. 46, no. 7, pp. 13–14, 7 2009.
- [5]. Akan OB, Fang J, and Akyildiz IF, "TP-planet: A reliable transport protocol for interplanetary Internet," *IEEE J. Sel. Areas Commun*, vol. 22, no. 2, pp. 348–361, 2 2004.
- [6]. De Cola T and Marchese M, "Performance analysis of data transfer protocols over space communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1200–1223, 10 2005.
- [7]. Bisio I and Marchese M, "Analytical expression and performance evaluation of TCP packet loss probability over geostationary satellite," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 232–234, 4 2004.
- [8]. Caini C, Cornice P, Firrincieli R, Livini M, and Lacamera D, "Analysis of TCP and DTN retransmission algorithms in presence of channel disruptions," in *Proc. 1st Int. Conf. Adv. Satellite Space Commun.*, 7 2009, pp. 174–179.
- [9]. Wang R, Wu X, Wang T, Liu X, and Zhou L, "TCP convergence layer-based operation of DTN for long-delay cislunar communications," *IEEE Syst. J.*, vol. 4, no. 3, pp. 385–395, 9 2010.
- [10]. Wang R, Modi B, Zhang Q, Hou J, Guo Q, and Yang M, "Use of a hybrid of DTN convergence layer adapters (CLAs) in interplanetary Internet," in *Proc. IEEE ICC*, 6 2012, pp. 3296–3300.
- [11]. Yu Q, Burleigh SC, Wang R, and Zhao K, "Performance modeling of licklider transmission protocol (LTP) in deep-space communication," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1609–1620, 7 2015.
- [12]. Yang Z et al., "Analytical characterization of Licklider transmission protocol (LTP) in cislunar communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 2019–2031, 7 2014.
- [13]. Wu H, Li Y, Jiao J, Cao B, and Zhang Q, "LTP asynchronous accelerated retransmission strategy for deep space communications," in *Proc. IEEE Int. Conf. Wireless Space Extreme Environ. (WiSEE)*, 9 2016, pp. 99–104.

- [14]. Lu H, Jiang F, Wu J, and Chen CW, "Performance improvement in DTNs by packet size optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 2987–3000, 10 2015.
- [15]. Bezirgiannidis N and Tsaoussidis V, "Packet size and DTN transport service: Evaluation on a dtn testbed," in *Proc. Int. Congr. Ultra Modern Telecommun. Control Syst.*, 10 2010, pp. 1198–1205.
- [16]. Bezirgiannidis N, Burleigh S, and Tsaoussidis V, "Delivery time estimation for space bundles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 3, pp. 1897–1910, 7 2013.
- [17]. Wei Z, Wang R, Zhang Q, and Hou J, "Aggregation of DTN bundles for channel asymmetric space communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 6 2012, pp. 5205–5209.
- [18]. Wang R, Reshamwala A, Zhang Q, Zhang Z, Guo Q, and Yang M, "The effect of 'window size' on throughput Perform. DTN lossy cislunar communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 6 2012, pp. 68–72.
- [19]. Zhao K, Wang R, Burleigh S, Qiu M, Sabbagh A, and Hu J, "Modeling memory-variation dynamics for the Licklider transmission protocol in deep-space communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 2510–2524, 10 2015.
- [20]. Wang R, Burleigh SC, Parikh P, Lin C-J, and Sun B, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 359–368, 4 2011, doi:10.1109/TNET.2010.2060733.
- [21]. Wang D, "Performance of licklider transmission protocol (LTP) in leo-satellite communications with link disruptions," in *Proc. IEEE 15th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI,CC)*, 8 2016, pp. 154–159.
- [22]. Farrell S and Cahill V, "Evaluating LTP-T: A DTN-friendly transport protocol," in *Proc. Int. Workshop Satellite Space Commun.*, 9 2007, pp. 178–181.
- [23]. Alessi N, Burleigh S, Caini C, and Cola TD, "LTP robustness enhancements to cope with high losses on space channels," in *Proc. 8th Adv. Satellite Multimedia Syst. Conf. 14th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, 9 2016, pp. 1–6.
- [24]. Shi L 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, 4 2017.
- [25]. Burleigh S, Ramadas M, and Farrell S, Licklider Transmission Protocol—Motivation, document RFC 5325, 9 2008.
- [26]. Ramadas M, Burleigh S, and Farrell S, Licklider Transmission Protocol—Specification, document RFC 5326, 9 2008.
- [27]. Durrett R, *Probability Models for DNA Sequence Evolution (Probability and Its Applications)*. New York, NY, USA: Springer, 2008, doi:10.1007/978-0-387-78168-6.
- [28]. Lent R, "INES: Network simulations on virtual environments," in *Proc. 1st Int. Conf. Simulation Tools Techn. Commun., Netw. Syst. (SIMUTools)*, Marseille, France, 2008.

Fig. 1.

Main cases involved in the transmission of a LTP block: (A) nominal case; (B) one or more DS losses excluding CP; (C) CP segment loss; (D) DS and RS loss; (E, F) RS or RA loss. A cross (“X”) indicates a segment loss. T is the block delivery time. Case (E) shows a second possible (implementation dependent) outcome—RS/RA with discontinuous lines.

Fig. 2.
Embedded Markov Chain that describes the stochastic process of a LTP session with epochs occurring at the beginning of each transmission round.

(a)

(b)

Fig. 3.

(a) Average number of visits to transient state k as calculated by simulation and with the approximated expression (17) for a session of 100 data segments and $p' = 0$, (b) Average number of steps to absorption obtained by simulation and expression (18) for a session of 100 segments.

(a)

(b)

Fig. 4.

Block delivery time vs. channel BER with segment lengths (a) 500 bytes and (b) 1000 bytes according to both the approximate model and event-driven simulation of the BP/LTP.

(a)

(b)

Fig. 5.

(a) Approximate model and simulation results for the average number of rounds per block vs. channel BER for different segment lengths, (b) RMSE of the proposed method compared to existing methods.

(a)

(b)

Fig. 6.

(a) Delivery time of a block of size 100 MB on a hypothetical mars–earth link with LTP as a function of the selected segment payload length, (b) optimal LTP segment length for a session under different channel conditions.

TABLE I

Summary of the Main Parameters Used in the Analysis

	Param.	Meaning
app	λ	Data generation rate (bps)
LTP	b	Average block size (bits)
	b^+	Maximum block aggregation length (bits)
	b^o	Maximum block Aggregation time (s)
	f	Avg. red-part fraction: f , avg. green-part fraction: $1 - f$
	h	LTP header length (bits)
	κ	Average number of rounds (transmission efforts)
	m	Segment payload length (bits)
	n	Number of red-part segments in the block ($n = \lambda b/m$)
	p	Data segment (DS) loss probability
	p'	Probability of timeout
	$p_c p_r$	CP segment/Report segment (RS) loss probability
frame	h_f	Frame header length (bits)
	L	Total frame length (bits)
	m^+	Maximum transmission unit (bits)
	p_f	Frame loss probability
	χ	Max. number of bit errors allowed for frame decoding
channel	d	Propagation delay (s)
	p_e	Bit loss probability, bit error rate (BER)
	r	Transmission rate (bps)
	w	Average offline and/or buffering time per block