Analysis of the Block Segmentation Method of the Licklider Transmission Protocol

Ricardo Lent(✉)

University of Houston, Houston, TX, USA
rlent@uh.edu

Abstract. Space communications are continuously challenged by extreme conditions that include large propagation delays, intermittent connectivity, and random losses. To combat these problems, the Licklider Transmission Protocol (LTP) splits data blocks into small segments that are radiated independently and retransmitted as needed, through a process that can be paused during long link disruptions. Given the extreme delays involved, the end performance of this protocol is driven by the number of transmission rounds needed to successfully deliver each block. LTP links are defined as overlays with one or more physical channels in the underlay, therefore with sections that may be on different administrative domains and experiencing different conditions. The question of how to select the length of the segments has received negligible attention and the use of improper values can easily lead to suboptimal performance. The segmentation process used by LTP is examined in this paper to determine the role that segmentation parameters and the conditions of the underlay have on the block delivery times. This goal is achieved through the definition of a basic model of LTP's transmission process that allows deriving the optimal segmentation parameter. Simulation results provide additional evidence of LTP's performance contrasting the results of the optimal segment length with fixed-length segments. The results provide a theoretical performance reference for practical parameter optimization methods.

Keywords: Delay tolerant networks · Deep space communications · Satellites · Licklider Transmission Protocol · Protocol optimization

1 Introduction

Space communications networks are made distinctive by the use of long-distance links, which both entail extreme one-way propagation light times (OWLT) and long down periods. For example, the OWLT for Earth-Moon communications is around 1.2 s whereas for the one for Earth-Mars is in the range of 4–24 min. Orbital mechanics also bring occultation (i.e., celestial bodies blocking communications), which can prevent any communication from a few minutes to

P. Gaj et al. (Eds.): CN 2020, CCIS 1231, pp. 14–26, 2020.
https://doi.org/10.1007/978-3-030-50719-0_2

several hours. Along with the Bundle Protocol (BP), the Licklider Transmission Protocol was developed specifically to address those issues. LTP defines overlay links with support for both best-effort and reliable data delivery. Because of the use of network abstraction, LTP links are built on top of another protocol as desired without constraint to any specific layer. Therefore, the underlay could be selected to be a link-layer protocol or a higher-layer protocol, such as TCP or UDP.

In the current state-of-practice, space radio channels and the link-layer are regularly tuned to improve their performance, e.g., by redesigning link budgets or through the use of error correction codes and buffer management techniques. However, the problem of how to optimize LTP-specific parameters has received relatively low attention in the past. The ION-DTN [1] implementation of LTP provides the means for the manual configuration of different protocol parameters, such as the maximum block size, number of import and export sessions (which controls the transmission concurrency), and the maximum segment size. However, limited automation for finding the optimal values for these parameters is available, requiring many times the use of human expertise.

In this paper, the segmentation method used by LTP is analyzed, which consists of dividing a data block into small chunks (i.e., segments) for transmission. In all current implementations, including ION-DTN, the maximum segment length is manually set usually matching the underlay's maximum transmission unit (MTU). The hypothesis is that the current approach for setting the segment length may not yield optimal performance for all situations as large segments tend to be more error-prone than smaller segments. Lost segments need to be retransmitted, which extends the delivery time for the block. However, the use of small segments may not be desirable either as they add header overhead both from LTP and the underlay. Therefore, the tradeoffs in the selection of the segment length require further examination.

The contributions of this paper are three-fold: (1) it analyzes LTP's block segmentation process, yielding a model that describes the role of the conditions of the underlay network and the selected protocol parameters to the overlay link efficiency, (2) it finds the theoretical ideal segment length that maximizes link efficiency and that leads to lower response times and higher throughout than achievable through the common practice of using fixed segment sizes, and (3) it provides the ideal LTP performance that serve as a reference for future works.

2 Related Works

A common criterium for determining the size of the data units in a network is the possibility of fragmentation. As stated in RFC 2488, which defines performance enhancements for satellite channels using standard network mechanisms, it is recommended "the use of the largest packet lengths that prevent fragmentation", which can be determined by the Path MTU Discovery (RFC 1191) mechanism. In addition, it recommends the use of forward error correction (FEC) to reduce the possibility of triggering congestion control actions when TCP is used.

Several works have suggested the practical link between the selected packet sizes and performance, in particular, considering challenged networks. For example, Basagni et al. studied the impact of the packet size selection in an underwater wireless sensor network [2] showing that the performance of the carrier sense multiple access (CSMA) and the distance-aware collision avoidance protocol (DACAP) is impacted by the packet size selection even with BER values as low as 10^{-6}.

Space networks are challenged networks that are especially exposed to the impact of packet losses due to the long propagation delay of the links. This observation has been well documented by numerous studies [3–6] that have focused on different aspects of LTP, such as flow control [7], the aggregation of bundles in blocks [8,9] and the impact of the selection of the convergence layer [10,11]. Recent works have focused on enhancing LTP for performance gains [12–14], for example through the use of a Reed-Solomon code [15] to reduce the segment loss probability.

An experimental work, conducted by Bezirgiannidis and Tsaoussidis [16], measured the effect that packet sizes have on LTP performance. Several works have looked into the properties of the radio channel and the formulation of the packet optimization problem (e.g., see [17,18]). Close work was carried out by Lu et al. who analyzed the approximated impact of packet sizes to LTP's performance and formulated a heuristic to find the optimal length [19]. Because of the use of a performance model, a drawback of their approach is that it requires knowledge of the channel state which may not be available given the overlay nature of LTP links and the possible lack of cross-layer information. In recent work, a cognitive networking approach to the dynamic selection of the optimal segment length that does not require precise knowledge of the underlay was also proposed [20].

3 Synopsis of the Licklider Transmission Protocol

The Licklider Transmission Protocol (LTP) [21,22] was introduced as a convergence layer protocol for the Bundle Protocol (BP) to support bundle transmissions over one or multiple links (as an overlay) that are expected to be disrupted for extended times. LTP receives and transmits *service data units* from and to BP (i.e., data bundles). As data arrives from BP, the sending LTP engine accumulates the data in a buffer to create a data block. Each data block is segmented and sent independently of each other to the receiving LTP engine. To this end, LTP uses lower layer protocols. Typically, CCSDS/AOS (Consultative Committee for Space Data Systems/Advanced Orbit System) for single-hop radio links and internet protocols (UDP, TCP, STCP) for overlays. The size of the data blocks is determined by two parameters, which limit the amount of memory reserved for each block and the filling time (typically, 1 s) respectively. Therefore, the actual size of any particular block is not necessarily fix and may include a whole bundle, part of a bundle, or even multiple bundles.

LTP supports both best-effort and reliable transmissions within the same block. The portion of the block devoted to best-effort is labeled the *green-part*, whereas the *red-part* is reserved for the reliable part. This division is possible because each block is segmented and each block transmission consists in sending a sequence of data segments (DS). This process is called a *session*. The size of a segment is commonly set to match the value of the maximum transmission unit (MTU) of lower-layer protocol(s), but the benefits of selecting smaller segments based on the context are suggested in this paper. The protocol does not put any particular limit on the size of the segments, so that segments may be smaller than the MTU or even larger (therefore, spanning multiple data-link frames).

Each session requires at least one transmission round. During that round all of the block's DS are sent. The last segment is labeled the *checkpoint segment* (CP) and carries the flag *end-of-red-part* (EORP). This flag tells the receiver to respond with a report segment (RS). The RS provides negative acknowledgment to the sender so that it can retransmit lost segments. Additional rounds can proceed until all of the segments are either successfully received or the maximum number of rounds is reached. In the latter case, the block is discarded. On reception of the final confirmation (RS) from the receiver, the sender transmits an acknowledge (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 available to LTP at both ends. From the standpoint of the sender, the session finishes as soon as the RA is transmitted. Figure 1 illustrates the process taking place as an overlay link.

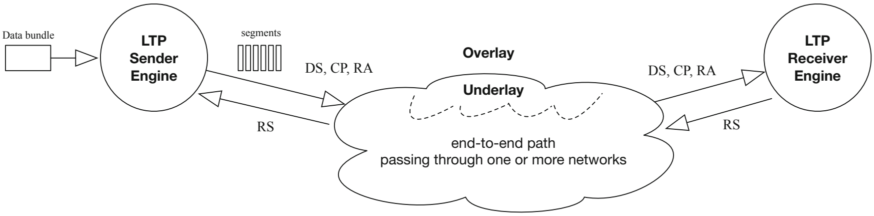


Fig. 1. An LTP link defines a network abstraction (overlay) that is not restricted to a single physical link, but that runs on top of an arbitrary underlay such as UDP/IP.

4 Analysis

To analyze the theoretical impact of the segmentation process, the study first looks at the nominal (i.e., error-free) overlay link efficiency and capacity and then extends the results to the lossy case. Capacity refers to the upper bound of the rate at which bundles can be reliably received.

The study considers the general case of application data being reliably sent over an overlay link by BP over LTP. The overlay is built over an underlay of Z channels, that is, the underlay path consists of Z physical links. In the discussion that follows, a *frame* is the name of the underlay protocol data unit.

Despite the usual association of that term to the link layer, no specific restrictions are assumed about the underlying protocol, so a *frame* could be a UDP datagram for instance or a CCSDS frame.

4.1 Nominal Bundle Capacity

Each bundle consists of at least two structures. One structure carries the payload, whereas the second, and additional structures if any, carries control information. Let us use B to represent the total bundle length and h_B the total length used by the control structures within the bundle. The mapping of bundles to LTP blocks depends on the amount of application data available to be sent, implementation specifics, and protocol configuration. It may result in blocks carrying single bundles, a fraction of a bundle, or multiple bundles. To model these alternatives, let use parameter a to indicate the *bundle-block aggregation factor*: $a = B/b$, where b is the block size. If $a = 1$, then one block carries exactly one bundle; if $a > 1$, multiple blocks ($\lceil a \rceil$ blocks) carry one bundle; and, if $a < 1$, one block aggregates different bundles. Furthermore, let m and h represent the segment payload length and header length respectively. To simplify the notation, let us assume that h includes the lengths of both the segment and link-layer frame so that the total segment length at the physical layer is $L = m + h$. Protocol extensions, such as security mechanisms extensions [23], involve the addition of extra control information and so, lead to a larger value for h. Each block requires the transmission of $n = b/m$ segments or frames, since it is assumed that each segment travels in one frame.

The nominal BP/LTP efficiency is the ratio of the maximum BP payload length to the total transmission length including overhead and data. To calculate the protocol efficiency, it can be noted that the total overhead of a bundle transmission over LTP is $h_B + anh$ given that each bundle introduces control overhead (h_B) and involves the transmission of a blocks of n segments. Each segment within each block adds a separate protocol overhead h. The nominal BP/LTP efficiency $\mathcal{E}_{nom}$ is, therefore:

$$\mathcal{E}_{nom} = \frac{B - h_B}{B + ahn} = \frac{1 - h_B/B}{1 + h/m} \tag{1}$$

The values of both h_B and h are comparatively fixed, and the value of B depends on the application requirements. The numerator in (1), $1 - h_B/B$, represents the bundle efficiency and varies between 0 and 1. Increasing the bundle overhead h_B decreases bundle efficiency linearly. Values of B of at least 10 times larger than h_B allow achieving a bundle efficiency of 0.9 or higher. The denominator in (1), $1+h/m$, is the frame overhead factor. The expression indicates that larger packet payload m helps to reduce the frame overhead and also protocol efficiency. It can also be observed that realistically, m is the only controllable variable that affects the nominal BP/LTP protocol efficiency. Another observation is that the block size b and the aggregation factor a do not impact nominal protocol efficiency.

The nominal link overlay capacity is the product of its efficiency and the end-to-end throughput r. With the assumption of negligible buffering delays in this work, the processing time of frames at the different sections of the overlay are only determined by the links' rates. Therefore, the overlay link's throughput is given by the slowest link of the underlay path. Furthermore, let us denote with v the overlay path availability that results due to link disruptions (e.g., as caused by occultation or link scheduling actions in the space communications environment). The nominal capacity θ^{+}_{nom} is given by: $\theta^{+}_{nom} = \mathcal{E}_{nom} v r$.

4.2 Reliable Bundle Efficiency and Overlay Link Capacity

Consider now the reliable transmission of bundles over a lossy overlay link that randomly drops frames with probability p. Because the overlay may consist of multiple sections, a number of these sections may implement forward error codes, such as Reed-Solomon coding or Turbo coding, that can help to mitigate frame drops. Also, additional packet encapsulation and multiplexing may occur prior to the physical layer transmission at each section. Parameter p then models the resulting end-to-end packet drop probability after considering any error mitigation technique used in all sections of the underlay. In detail, let b_i denote the bit error rate (BER) of the i-th channel of the underlay. For a frame of size L with independent bit errors, the packet error rate p_i for section i, $i = 1, \ldots Z$, where Z is the underlay path length, is given by the expression:

$$p_i = 1 - (1 - b_i)^L = 1 - e^{L.log(1-b_i)} \tag{2}$$

Because frames could be dropped at any section of the underlay, parameter p is then given by the expression:

$$p = 1 - \prod_{i=1}^{Z} (1 - p_i) \tag{3}$$

It is relevant to emphasize that in practice frames may be also dropped due to congestion. The model in this paper only considers channel losses and a negligible probability of buffer overflow. Assuming that each segment travels on a separate frame, the packet loss probability p will affect the transmission capacity of the overlay θ^{+}_{s} as follows:

$$\theta^{+}_{s} = (1 - p) v r / L \tag{4}$$

given that the maximum segment throughput of r/L is also limited by the overlay link availability v and only the fraction $(1-p)$ of the segments will not be rejected on average due to errors. Because each LTP block consists of n segments, the maximum reliable block transmission capacity (θ^{+}_{b}) becomes:

$$\theta^{+}_{b} = \theta^{+}_{s} / n. \tag{5}$$

Despite LTP implements reliable block transmission, a block may still be dropped after a certain number of unsuccessful delivery attempts. It is generally beneficial to decide first the value of the target block loss rate (α) and then calculate that maximum number of rounds that can achieve such a level [24]. It is considered that α is known as is given as an upper bound by design and that $\alpha \ll 1$, which is typical and allows ignoring the impact of the maximum number of rounds to the average value (κ).

If the bundle fits within one LTP block, then the probability of losing the bundle is just α. However, if the bundle involves several block transmissions, then at least one block loss causes the entire bundle rejection. A bundle is not lost with probability $(1-\alpha)^A$, where $A = max\{1, a\}$, where a is the bundle-block aggregation factor previously defined. The maximum reliable bundle capacity (θ_B^+) is then:

$$\theta_B^+ = (1-\alpha)^A \theta_b^+ / a. \tag{6}$$

Observing that each bundle carries $(B - h_B)$ of useful data, and using (4), (5) and (6), it can be determined the reliable capacity of BP/LTP as:

$$\theta^+ = (B - h_B)\theta_B^+ = \mathcal{E} r v / a \tag{7}$$

where

$$\mathcal{E} = \frac{1 - h_B/B}{1 + h/m}(1-\alpha)^A(1-p) = \mathcal{E}_{nom}(1-\alpha)^A(1-p). \tag{8}$$

The reliable efficiency $\mathcal{E}$ is equivalent to the nominal efficiency subject to not losing a segment nor a bundle. Unlike the nominal efficiency case, parameter m not only impacts the segment overhead factor in the reliable efficiency, but also the segment loss probability p (3). Expression (8) also suggests that the transmission of a bundle using multiple blocks decreases the reliable efficiency.

Figure 2 (a) compares the nominal BP/LTP efficiency with the reliable BP/LTP efficiency for a range of values of the segment payload length m and under different BER conditions. The model parameters include 100 kB bundles with 100 B control overhead, an LTP header length of 12 B, one block per bundle ($a = 1$), and a block loss rate of 10^{-6}. At low packet loss rates, the nominal and reliable overlay link efficiencies are very similar. The situation changes at high packet loss rates in particular with large segments.

4.3 Optimal Payload Length

As previously indicated (7), only the efficiency factor that appears in the overlay link capacity depends on the segment payload length m. The overlay link throughput (r), overlay availability (v), and bundle-block aggregation factor (a) are insensitive to the choice of the segment payload length. The optimal value of the segment payload length m^* yields the largest $\mathcal{E}$, which can be found by solving:

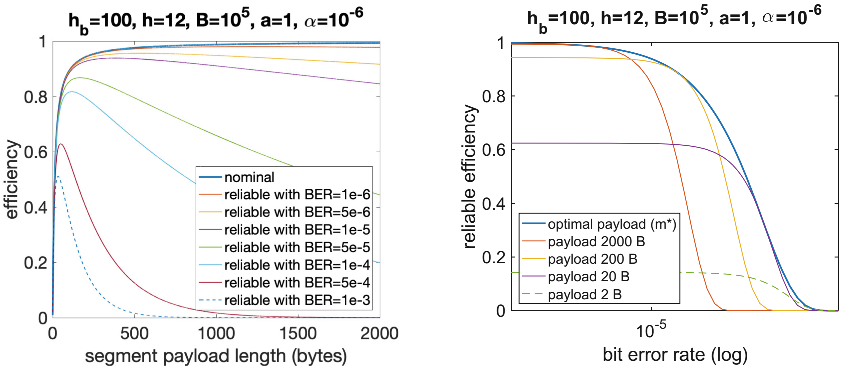
$$\begin{aligned} \underset{m}{\text{maximize}} \quad & \mathcal{E}(m) \\ \text{subject to} \quad & 0 < m \le B, M \\ & m \in \mathbb{Z}. \end{aligned}$$

where M is the smallest payload-length that is supported along the path of the underlay network. The single-section network case (i.e., single-hop, $b = b_1$) leads to a simple expression that helps to illustrate the tradeoffs involved in the segment length selection. It can be found by temporarily ignoring the two constraints and finding the horizontal tangent with $\mathcal{E}' = 0$:

$$m^+ = \frac{h}{2}\left(1 + \sqrt{1 - \frac{4}{h\,log(1-b)}}\right). \tag{9}$$

The constraints can then be applied using: $m^* = max\{1, min\{\lfloor m^+ \rfloor, B, M\}\}$. Expression (9) reinforces the notion that larger segments are generally better suited for low BER with the opposite case otherwise.

Figure 2 (b) depicts the reliable efficiency achieved by the optimal payload under different BER values, compared to fixed payloads of 2, 20, 200, and 2000 bytes. The case corresponds to a bundle size of 10^5, single bundle carrying blocks, and bundle and frame overhead of 100 and 12 bytes respectively. The probability of block loss was fixed at 10^{-6}. The results illustrate the tradeoffs between header overhead and segment drops involved in the selection of the LTP segment length.



(a) (b)

Fig. 2. (a) Nominal vs. reliable BP/LTP protocol efficiency, and (b) efficiency achieved with fixed segments of various lengths (i.e., LTP) and with the optimal segmentation.

5 Simulation Results

To verify the performance achievable by the optimal segmentation, a discrete event simulator of BP/LTP was used. The performance was measured in terms of the block delivery time, response time, and bundle throughput. The simulator keeps track of the timing of all the packet buffering and transmission events, in addition to the packet drops, of a packet flow that is being carried by a single

256-kbps wireless channel with given one-way light time (OWLT) and bit error rate (BER). The accuracy of the simulator has been verified previously [24,25].

The OWLT and BER channel values were used as experimental factors. It is assumed that the link was always available ($v = 1$). Statistics of the transmission of 1000 bundles of size 100 kB were collected to characterize LTP performance. Identical parameters to the theoretical model were assumed in the simulation: a bundle fits exactly one block ($a = 1$) and the block control overhead was fixed to 100 bytes. In addition, each segment carries a header of 12 bytes and RS/RA (report acknowledgement) are 20 and 7 bytes long respectively.

5.1 Optimal Segment Size

Figure 3 (a) depicts the average segment length as determined by expression 9. The independent parameter is the BER value of the single channel used in the simulations. As a reference, the results obtained with fixed segment lengths of 500, 1000, and 1400 B have been included. These values are within the typical range used in practice, for example, RFC 879 states that the default maximum segment size (MSS) for TCP is 536 B and the general recommendation for LTP in the ION-DTN implementation is the use of 1400 B. The evaluation included an intermediate segment length between these two extremes. The optimal segment length was constrained to the range 10–1400, which explains the constant optimal value for BER values $<10^{-6}$.

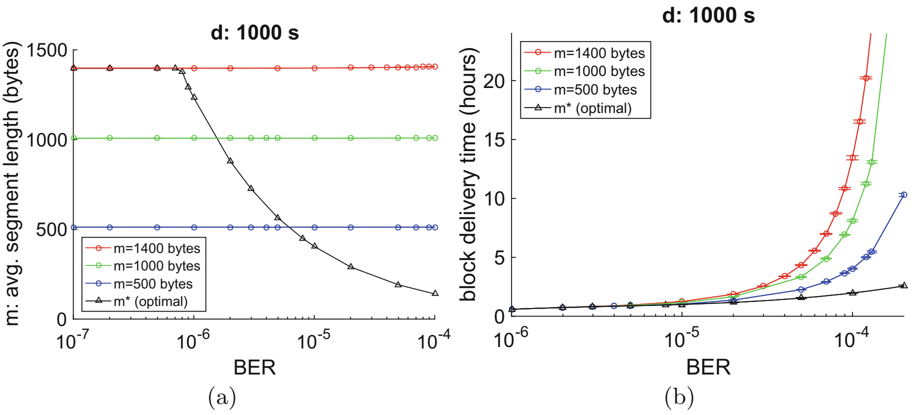
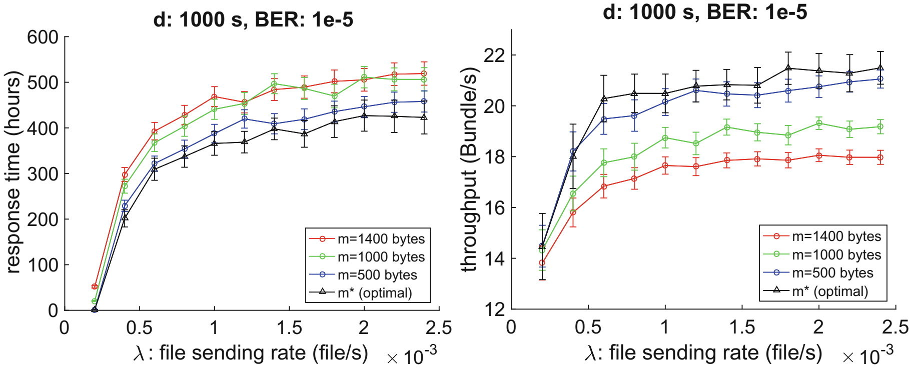


Fig. 3. (a) Optimal segment lengths and (b) block delivery times.

Figure 3 (b) depicts the average bundle delivery times obtained with the optimal segmentation and the fixed segment lengths. The performance advantage of adjusting the segment length become apparent for BER values higher than 10^{-5}. In the simulations, no error-correcting code was assumed. However, the coding gain of real channels is expected to simply shift the results along the horizontal axis.

5.2 Average Block Response Time and Throughput

Unlike the block delivery time (i.e., service time), the block response time is affected by the block sending rate and buffering. Higher sending rates increase the chances of buffer congestion extending response times. The effect is depicted in Fig. 4 (a) with a BER of 10^{-5} and for a range of values for the sending rate up to 2.3×10^{-3} bundle/s. It is worth noting that the values that are shown in the chart are not steady-state averages, but the average response time for 1,000 bundle transmissions at the selected rate. The benefits become particularly significant, measured in the range of hours for the 1,000 s link, as bundles are sent at higher rates. The optimal segmentation achieves up to 20–30% higher throughput than with the use of fixed segment lengths as depicted in Fig. 4 (b).



(a) (b)

Fig. 4. Bundle response time (a) and throughput (b) over a single channel with a BER value of 10^{-5} and one-way propagation delay (d) of 1,000 s as a function of the bundle sending rate. The values were calculated for the first 10^3 bundles (not steady state).

5.3 Impact of the Propagation Delay

While the propagation delay does not affect the average number of rounds required to delivery a block by LTP, the time required for each round is a function of the one-way propagation delay of the channel. The relationship between block delivery time and propagation delay is linear as it can be observed in Fig. 5 for BER values of 10^{-5} and 10^{-4}. This evaluation covers a wide range of OWLT, from terrestrial to values associated to links to the edge of the solar system.

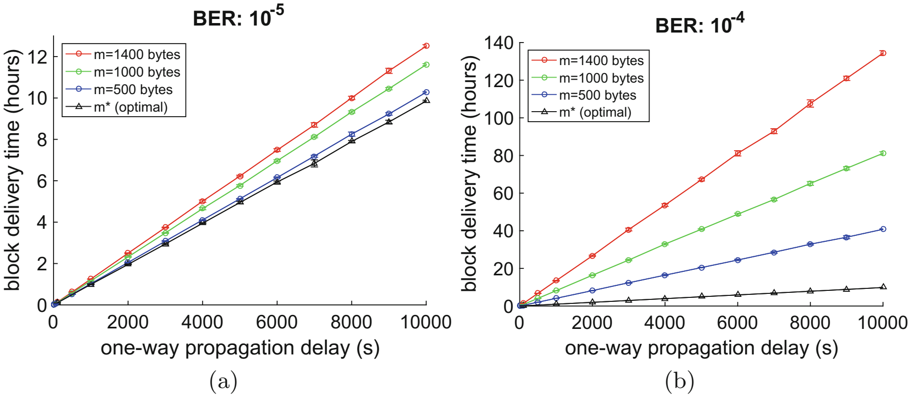


Fig. 5. Block delivery time vs. one-way propagation delay with channel BER values of (a) 10^{-5} and (b) 10^{-4}.

6 Conclusion

In this paper, the connection between bundle delivery performance and both the overlay link properties and the choice of the segmentation parameter was analyzed. A model of the overlay link efficiency was proposed and used to find the optimal segment length. Contrary to the default practice of setting the length according to the maximum transmission unit of the underlay, the results show that such practice may lead to suboptimal delivery times when the end-to-end packet loss ratio is high. Given that end-to-end communication conditions can change at any time, it can be inferred that an online method for defining LTP parameters should be employed rather than keeping this parameter constant as commonly done in practice today. Moreover, the overlay nature of LTP links may complicate the optimization as multiple physical-layer channels may be involved in the underlay negating the possible benefits of static techniques that try to map BER measurements to segment lengths. Nevertheless, it is expected that the results of this study will help to define a performance reference for future practical methods for LTP parameter optimization.

Acknowledgment. The author would like to thank Gilbert Clark at NASA Glenn Research Center for his useful comments on this research. This work was supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program.

References

1. The Interplanetary Overlay Network (ION) software distribution: ION-DTN. https://sourceforge.net/projects/ion-dtn. Accessed 01 Apr 2018
2. Basagni, S., Petrioli, C., Petroccia, R., Stojanovic, M.: Optimized packet size selection in underwater wireless sensor network communications. IEEE J. Ocean. Eng. **37**, 321–337 (2012)

3. Bisio, I., Marchese, M.: Analytical expression and performance evaluation of TCP packet loss probability over geostationary satellite. IEEE Commun. Lett. **8**(4), 232–234 (2004)
4. Caini, C., Cornice, P., Firrincieli, R., Livini, M., Lacamera, D.: Analysis of TCP and DTN retransmission algorithms in presence of channel disruptions. In: 2009 First International Conference on Advances in Satellite and Space Communications, pp. 174–179, July 2009
5. Wang, R., Burleigh, S.C., Parikh, P., Lin, C.J., Sun, B.: Licklider Transmission Protocol (LTP)-based DTN for cislunar communications. IEEE/ACM Trans. Netw. **19**(2), 359–368 (2011). http://dx.doi.org/10.1109/TNET.2010.2060733
6. Wang, D.: Performance of Licklider Transmission Protocol (LTP) in LEO-satellite communications with link disruptions. In: 2016 IEEE 15th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), pp. 154–159, August 2016
7. Wang, R., Reshamwala, A., Zhang, Q., Zhang, Z., Guo, Q., Yang, M.: The effect of "window size" on throughput performance of DTN in lossy cislunar communications. In: 2012 IEEE International Conference on Communications (ICC), pp. 68–72, June 2012
8. Bezirgiannidis, N., Burleigh, S., Tsaoussidis, V.: Delivery time estimation for space bundles. IEEE Trans. Aerosp. Electron. Syst. **49**(3), 1897–1910 (2013)
9. Wei, Z., Wang, R., Zhang, Q., Hou, J.: Aggregation of DTN bundles for channel asymmetric space communications. In: 2012 IEEE International Conference on Communications (ICC), pp. 5205–5209, June 2012
10. Wang, R., Modi, B., Zhang, Q., Hou, J., Guo, Q., Yang, M.: Use of a hybrid of DTN convergence layer adapters (CLAs) in interplanetary Internet. In: ICC, pp. 3296–3300. IEEE (2012)
11. Wang, R., Wu, X., Wang, T., Liu, X., Zhou, L.: TCP convergence layer-based operation of DTN for long-delay cislunar communications. IEEE Syst. J. **4**(3), 385–395 (2010)
12. Farrell, S., Cahill, V.: Evaluating ltp-t: a DTN-friendly transport protocol. In: 2007 International Workshop on Satellite and Space Communications, pp. 178–181, September 2007
13. Alessi, N., Burleigh, S.C., Caini, C., de Cola, T.: LTP robustness enhancements to cope with high losses on space channels. In: 8th Advanced Satellite Multimedia Systems Conference and the 14th Signal Processing for Space Communications Workshop, ASMS/SPSC 2016, Palma de Mallorca, Spain, 5–7 September 2016, pp. 1–6 (2016)
14. Iannicca, D., Hylton, A., Ishac, J.: A performance evaluation of NACK-oriented protocols as the foundation of reliable Delay-Tolerant Networking convergence layers. NASA technical memorandum (2012)
15. Shi, L., et al.: Integration of reed-solomon codes to Licklider Transmission Protocol (LTP) for space DTN. IEEE Aerosp. Electron. Syst. Mag. **32**(4), 48–55 (2017)
16. Bezirgiannidis, N., Tsaoussidis, V.: Packet size and DTN transport service: evaluation on a DTN testbed. In: International Congress on Ultra Modern Telecommunications and Control Systems, pp. 1198–1205, October 2010
17. Carek, D.A.: Packet-based protocol efficiency for wireless communications. JACIC **2**, 238–251 (2005)
18. Korhonen, J., Wang, Y.: Effect of packet size on loss rate and delay in wireless links. In: IEEE Wireless Communications and Networking Conference, WCNC, vol. 3, pp. 1608–1613, April 2005

19. Lu, H., Jiang, F., Wu, J., Chen, C.W.: Performance improvement in DTNs by packet size optimization. IEEE Trans. Aerosp. Electron. Syst. **51**(4), 2987–3000 (2015)
20. Lent, R.: A cognitive networking technique for LTP segmentation. In: The International Wireless Communications and Mobile Computing Conference, June 2020
21. Burleigh, S., Ramadas, M., Farrell, S.: Licklider Transmission Protocol - motivation. RFC 5325, RFC Editor, September 2008
22. Ramadas, M., Burleigh, S., Farrell, S.: Licklider Transmission Protocol - specification. RFC 5326, RFC Editor, September 2008
23. Farrell, S., Ramadas, M., Burleigh, S.: Licklider Transmission Protocol - security extensions. RFC 5327, RFC Editor, September 2008
24. Lent, R.: Regulating the block loss ratio of the Licklider Transmission Protocol. In: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6, September 2018
25. Lent, R.: Analysis of bundle throughput over LTP. In: 2018 IEEE 43rd Conference on Local Computer Networks (LCN). pp. 271–274 (October 2018)