SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2024, Vol. 67, Iss. 9, 192303:1–192303:16
https://doi.org/10.1007/s11432-024-4091-6

Data delivery delay and cross-layer packet size analysis for reliable transmission of Licklider transmission protocol in space networks

Guannan YANG[1], Ruhai WANG[2*], Kanglian ZHAO[3], Wenfeng LI[3] & Dong YAN[4]

[1]*School of Information Engineering, Nanjing University of Finance and Economics, Nanjing* 210023*, China;*
[2]*Drayer Department of Electrical and Computer Engineering, Lamar University, Beaumont* TX 77710*, USA;*
[3]*School of Electronic Science and Engineering, Nanjing University, Nanjing* 210093*, China;*
[4]*Beijing Aircraft Overall Design Department, Beijing* 100094*, China*

Received 25 February 2024/Revised 4 May 2024/Accepted 4 July 2024/Published online 16 August 2024

Abstract Delay/disruption tolerant networking (DTN) is proposed as a networking architecture to overcome challenging space communication characteristics for reliable data transmission service in presence of long propagation delays and/or lengthy link disruptions. Bundle protocol (BP) and Licklider Transmission Protocol (LTP) are the main key technologies for DTN. LTP red transmission offers a reliable transmission mechanism for space networks. One of the key metrics used to measure the performance of LTP in space applications is the end-to-end data delivery delay, which is influenced by factors such as the quality of spatial channels and the size of cross-layer packets. In this paper, an end-to-end reliable data delivery delay model of LTP red transmission is proposed using a roulette wheel algorithm, and the roulette wheel algorithm is more in line with the typical random characteristics in space networks. The proposed models are validated through real data transmission experiments on a semi-physical testing platform. Furthermore, the impact of cross-layer packet size on the performance of LTP reliable transmission is analyzed, with a focus on bundle size, block size, and segment size. The analysis and study results presented in this paper offer valuable contributions towards enhancing the reliability of LTP transmission in space communication scenarios.

Keywords Licklider transmission protocol, DTN, bundle protocol, cross-layer packets size, space communication

1 Introduction

A lot of work has been done in space communication with applications from low-orbit satellite (LEO) to deep space and interplanetary networks [1–3], which has promoted the development of space networking protocols/architectures [4–8]. Through the implementation of a series of space network experiments conducted under the guidance of NASA's Jet Propulsion Lab (JPL), the effectiveness of delay/disruption tolerant networking (DTN) structure is proven [9]. DTN was introduced as an internetworking architecture to ensure reliable delivery in challenging communication and networking environments. These environments are often characterized by an extremely long propagation delay, data loss or corruption, frequent and prolonged link disruptions, and asymmetric channel rates. Traditional networking protocols like TCP/IP often struggle to function effectively. Future manned deep space exploration and space flight missions are typical application scenarios of DTN technology, especially for reliable transmission in space communication. An in-depth study of DTN key technologies is important to understand the operation and performance of DTN. According to [9], the operation and reliable delivery of DTN rely on the bundle protocol (BP) and underlying convergence layer (CL) transport protocol [10]. The Licklider Transmission Protocol (LTP) [11, 12] is a widely-used CL protocol proposed by NASA to provide reliable and efficient transmission services that have been standardized and deployed in space missions [13]. In [14], LTP was applied to the deep-space image transmission system, effectively improving transmission efficiency. The transmission performance of LTP has attracted much attention, which is intuitively reflected in the

* Corresponding author (email: rwang@lamar.edu)

© Science China Press 2024

data delivery delay largely influenced by the characteristics of the transmission link. Furthermore, the cross-layer packet size also significantly affects the transmission performance of LTP. However, the standardization of LTP does not give detailed guidance on cross-layer packet sizes, which inspired this study. The cross-layer packet size in this study refers to the protocol data unit that interacts with LTP in the DTN architecture and the protocol data unit size of the LTP itself, mainly including bundle, block, and segment.

At present, there are some research results on LTP. Wang et al. [15–19] have conducted significant research on the development and analysis of protocols for cislunar systems and the Solar System Internet (SSI). In [20], an optimal retransmission timeout interval for BP to achieve the best goodput performance is modeled. Evaluation results and performance analysis can be found in the existing literature on LTP research [21, 22]. In [6], linear network coding is added to the LTP protocol to enhance transmission efficiency. In [23], the effect of queuing delay on transmission performance of LTP was analyzed. In [24], the impact of link interruption characteristics in space communication on LTP reliable transmission performance was analyzed. An enhanced LTP, named a multicolor LTP, was presented in [25]. The multicolor LTP always notifies the sender whether the data has been successfully delivered. In [26], the study presents an aggressive and proactive LTP control signal handling mechanism that can reduce the latency of LTP session closing time. In terms of research on the impact of cross layer packet size, Ref. [27] presented the influence of LTP block size on storage performance with theoretical models and simulations. The minimum number of bundles aggregated by the LTP block and the optimal aggregation were analyzed in [28] to maximize transmission efficiency while avoiding ACK (Acknowledgement) transmission delay in uplink. However, how channel characteristics restrict the block size was not explained. In [29], the theoretical models for packet size and block size were established, and the optimization scheme of packet size was proposed to maximize goodput performance. But the convergence of LTP was not considered in the theoretical model, and it is default that each block only encapsulates one bundle. In [30], the paper only focuses on the size of bundles and segments, without considering the impact of block size. The block size and segment size of LTP were modeled in [31], with the Markov chain used to analyze the delivery delay of block in LTP reliable transmission. However, Refs. [30, 31] did not consider retransmission limitations of data and control information, and only simulated through a designed simulator, lacking real data flow verification. In [32], a Q-learning method was used to adaptively determine the segment size. However, only the impact of segment size on LTP reliable transmission performance was concerned, and the impact of bundle and block size was ignored. In [33], the cross-layer BP/LTP block size in space vehicle communications is studied. However, a statistical average method was adopted for most of aforementioned performance analyses of LTP.

In this paper, an end-to-end reliable data delivery delay model of LTP red transmission is proposed using the roulette wheel algorithm. The roulette wheel algorithm is a method used for selecting an item from a set with probabilities proportional to specified weights [34, 35]. The roulette wheel algorithm can be a good model for space networks because it reflects the random properties of data transmission and loss in the space communication environment. Just as the algorithm randomly selects items with different probabilities, data packets in space networks can experience varying levels of probability for loss or corruption during transmission due to factors like signal interference, atmospheric conditions, or hardware malfunctions. By using the roulette wheel algorithm as a model, the likelihood of data loss in space networks can be simulated and analyzed more accurately which helps to optimize communication protocols and network design for improved reliability and efficiency. Therefore, compared to the widely adopted statistical average method, the roulette wheel algorithm is more in line with the random characteristics that are typical in space networks. In addition, the retransmission limitations of data and control information were considered in our model, which is more in line with the protocol mechanisms overlooked in existing studies. The impact of cross-layer packet size on the performance of LTP reliable transmission is also analyzed in this study.

The remainder of this paper is organized as follows. Second 2 proposes the end-to-end reliable transmission delivery delay model for LTP in space applications based on the LTP reliable transmission mechanism. The model validation and performance analysis are presented in Section 3. A summary and conclusion are provided at the end.

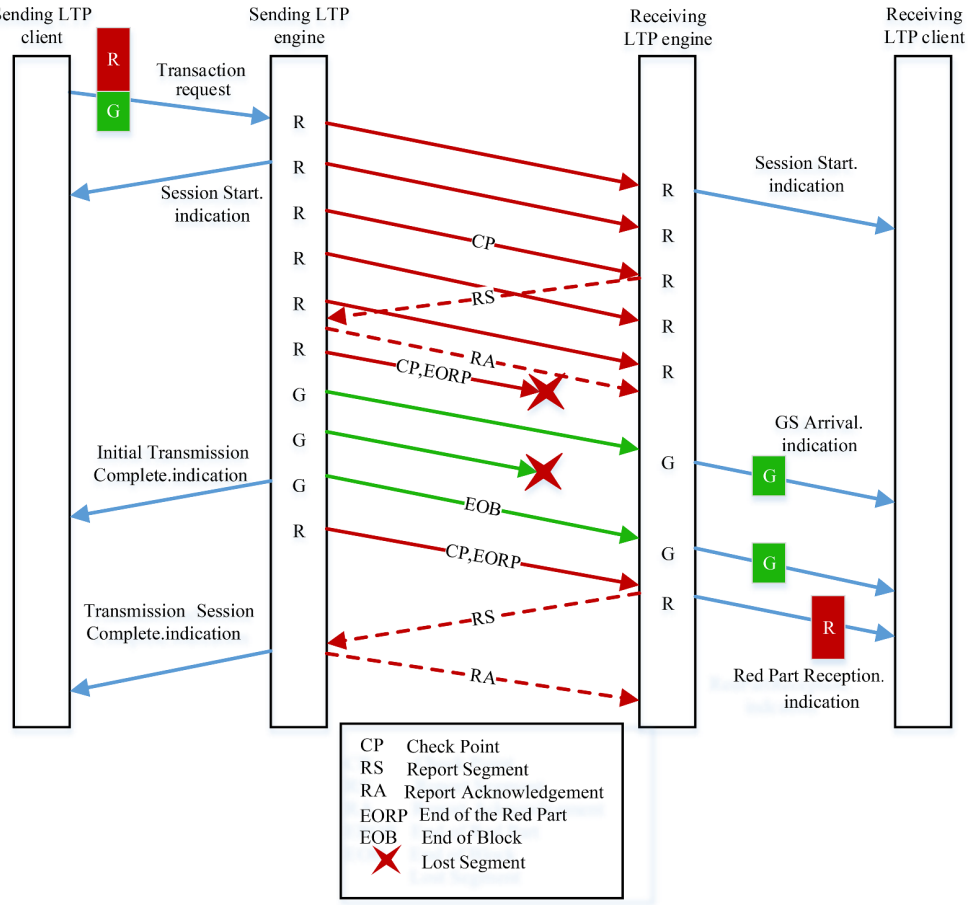

Figure 1 (Color online) LTP data transmission interaction process [13].

2 Delivery delay modeling for LTP reliable transmission

In DTN protocol stack [9], BP encapsulates some application data in a certain number of bundles and transmits these bundles as "service data units" to CL for transmission. Like TCP, LTP reliable transmission is achieved by retransmitting lost data. Figure 1 shows the encapsulation and de-encapsulation process of different layer protocol data units (PDUs) when a node sends and receives data.

The interaction process of LTP is shown in Figure 1, where the "red part" ensures reliable transmission through confirmation and retransmission, and the subsequent "green part" does not guarantee transmission reliability, that is, LTP protocol uses reliable transmission for red data and best-effort for green data. Therefore, in practical applications, if it is necessary to ensure reliable transmission of the entire data block, the length of the green part should be set to 0. Similarly, the entire data block can also be set as green data. In Figure 1, the last segment of the red part is identified as EORP (end of the red part) and CP (check point) to trigger the receiver for feedback. The end of the entire data block is identified by EOB (end of block). During the interaction process, when the receiver receives the CP segment, it will immediately respond with the confirmation report RS (report segment) to report the delivery status of the "red part", and each RS corresponds to one CP. After receiving RS, the sender immediately responds with a confirmation report RA (report acknowledgment). It is worth noting that when sending CP or RS control segments, both ends of the communication will start their respective local retransmission timers, in order to automatically retransmit if no corresponding confirmation is received within the timer limit. In terms of cross layer interaction, LTP encapsulates BP's protocol data unit bundle as an LTP block. By default, a bundle is encapsulated as a block. In the LTP mechanism, the confirmation of red data is achieved through the confirmation of blocks, that is, one confirmation for each block. There are also studies indicating that multiple bundles can be aggregated into a single block to limit the number of confirmations and reduce upstream data pressure in asymmetric links. The delivery process of each block is called a session, and each session is equivalent to an output queue. The current block in the session

Table 1 Notations for modeling process

Symbol	Definition
$T_{\text{deli\_file}}$	Predicted total file delivery delay
$T_{\text{prop\_file}}$	Predicted file delivery delay with transmission delay and effect of link disruption events excluded
$T_{\text{trans\_file}}$	Total file transmission delay in transmitting all the segments of the transmitted file, including all the retransmissions
$N_{\text{max\_exp\_session}}$	Maximum number of sessions
$L_{\text{max\_exp\_data}}$	Maximum amount of data being transmitted
$\overline{L_{\text{est\_exp\_block}}}$	Estimated output block length
T_{OWL}	The one-way light time
$R_{\text{est\_red}}$	Estimated "red data" transmission rate in bytes
$\overline{L_{\text{est\_agg\_block}}}$	Expected output block size
$L_{\text{ser\_data\_unit}}$	The service data unit size
L_{bundle}	The bundle size in bytes excluding the overhead of bundle
$L_{\text{head\_bundle}}$	The length of BP layer overhead
$L_{\text{agg\_size\_limit}}$	The block aggregation size threshold
$T_{\text{agg\_time\_limit}}$	The aggregation time limit
L_{file}	The file size in bytes
N_{block}	The number of blocks at LTP
N_{seg}	The number of segments for each block
P_{seg}	Transmission error probability of a segment
P	Bit-error-rate (BER) of the data transmission
L_{TH}	Total size of the PDU overhead from the LTP segment to the link frame
L_{seg}	Segment size that a block is fragmented
$N_{\text{max\_num\_RS}}$	The maximum number of Report Segments per block
$N_{\text{max\_CP\_retran}}$	The maximum number of retransmissions per CP
$N_{\text{max\_RS\_retran}}$	The maximum number of retransmissions per RS
L_{RS}	Length of an RS segment
L_{block}	Length of an LTP data block in bytes

can be sent only after all the previous blocks in the queue are correctly delivered. It is worth noting that LTP permits any number of sessions to be in progress concurrently, but only one session is set up in our model for simplicity. Then LTP divides each block into several LTP data segments according to the maximum transmission unit (MTU) of the link layer and encapsulates them into frames for transmission through the link layer.

In this section, analytical models are presented for LTP reliable transmission in space communications. The main task is to predict file delivery delay with data loss rate during file transmission. As done in the previous work in [24], the BP/LTP protocol stack architecture used in this study is configured with 100% LTP "red data", and the BP custody transfer option is disabled. LTP red transmission is one of the widely adopted DTN protocol configurations aimed at providing reliable transmission over unreliable space communication channels. Table 1 shows the symbols used in the models.

The total delivery delay is mainly composed of transmission delay and propagation delay ignoring protocol processing delay and queuing delay, which can be expressed as

$$T_{\text{deli\_file}} = T_{\text{prop\_file}} + T_{\text{trans\_file}}. \tag{1}$$

The maximum number of sessions ($N_{\text{max\_exp\_session}}$) should depend on the maximum amount of data ($L_{\text{max\_exp\_data}}$) being transmitted and the estimated output block length ($\overline{L_{\text{est\_exp\_block}}}$) between the sender and the receiver to avoid congestion, which can be formulated as follows:

$$N_{\text{max\_exp\_session}} = \frac{L_{\text{max\_exp\_data}}}{\overline{L_{\text{est\_exp\_block}}}}, \tag{2}$$

$$L_{\text{max\_exp\_data}} = 2 \times T_{\text{OWL}} \times R_{\text{est\_red}}/8, \tag{3}$$

where $R_{\text{est\_red}}$ is "red data" transmission rate in bits. For simplicity, only one session is used in the model, i.e., $N_{\text{max\_exp\_session}} = 1$ and $\overline{L_{\text{est\_exp\_block}}} = L_{\text{max\_exp\_data}}$. In addition, $\overline{L_{\text{est\_exp\_block}}}$ is limited by the size of the upper layer service unit, such as bundle size in BP. The estimated output block size

($\overline{L_{\text{est\_exp\_block}}}$) should be directly equal to the size of the upper layer PDU when this PDU is greater than the expected output block size ($\overline{L_{\text{exp\_agg\_block}}}$). Otherwise, it is determined by the expected output block size and equal to an integer multiple of the upper layer PDU size. Therefore, it can be expressed as

$$\overline{L_{\text{est\_exp\_block}}} = \lceil \max\{L_{\text{ser\_data\_unit}}, \overline{L_{\text{exp\_agg\_block}}}\}/L_{\text{ser\_data\_unit}} \rceil \times L_{\text{ser\_data\_unit}}, \tag{4}$$

where $L_{\text{ser\_data\_unit}}$ is the service data unit size. Because BP is adopted at the upper layer by default, $L_{\text{ser\_data\_unit}}$ is equal to the bundle size plus the BP overhead, i.e.,

$$L_{\text{ser\_data\_unit}} = L_{\text{bundle}} + L_{\text{head\_bundle}}. \tag{5}$$

It is worth noting that the expected block size ($\overline{L_{\text{exp\_agg\_block}}}$) is affected by the aggregation size threshold ($L_{\text{agg\_size\_limit}}$) and the aggregation time limit ($T_{\text{agg\_time\_limit}}$) of the block. In other words, once the aggregation size of the block reaches the threshold $L_{\text{agg\_size\_limit}}$ or the aggregation time reaches $T_{\text{agg\_time\_limit}}$, the block is immediately divided to form a series of data segments, which are delivered to the lower layer for transmission. Therefore, $L_{\text{agg\_size\_limit}}$ and $T_{\text{agg\_time\_limit}}$ determine the starting time of each block to be divided, and the relationship can be expressed as

$$\overline{L_{\text{exp\_agg\_block}}} = \min\{L_{\text{agg\_size\_limit}}, R_{\text{est\_agg\_red}} \times T_{\text{agg\_time\_limit}}\}. \tag{6}$$

Let L_{file} be the file length or, simply, file size. Considering that the adopted protocol stack architecture is BP/LTP, the number of blocks at LTP, N_{block}, and the number of segments for each block, N_{segment}, required for delivery of an entire file should be respectively formulated as

$$N_{\text{block}} = \frac{L_{\text{file}} \times (L_{\text{bundle}} + L_{\text{head\_bundle}})}{L_{\text{bundle}} \times \overline{L_{\text{est\_exp\_block}}}}, \tag{7}$$

$$N_{\text{seg}} = \left\lceil \frac{\overline{L_{\text{est\_exp\_block}}}}{L_{\text{seg}}} \right\rceil. \tag{8}$$

With respect to the quality of the transmission channel, let P_{seg} be the transmission error probability of a segment. Let the "effective net channel bit-error-rate (BER)" be P. Then, the probability that a given transmitted segment will suffer transmission error can be expressed as

$$P_{\text{seg}} = 1 - (1 - P)^{(L_{\text{seg}} + L_{\text{TH}})}, \tag{9}$$

where L_{TH} is the total overhead from the convergence layer LTP to the link layer. Although LTP "red" transmission provides reliability by retransmitting lost segments, it is not an endless retransmission until correct delivery. In the implementation mechanism of LTP, the number of report segments for each block is limited, excluding retransmission reports, and the maximum value of reception claim count is 20 in each report segment (i.e., RS). Therefore, this limits the number of times each block can be sent. In addition, there is also a constraint on the maximum number of retransmissions for control information. The maximum number of report segments per block can be formulated as

$$N_{\text{max\_red\_retran}} = 2 + \sum_{i=1}^{m} \left(\left\lfloor \frac{N \times (P')^i}{19} \right\rfloor + \left\lceil \frac{N \times (P')^i \% 19}{19} \right\rceil \right), \tag{10}$$

where m, N, and P' are as written as

$$m = \left\lceil \log_{P_{\text{loss}}} \left(\frac{1}{N_{\text{seg}}} \right) \right\rceil,$$

$$N = \max \left\{ 1, \left\lfloor \frac{\overline{L_{\text{est\_exp\_block}}}}{L_{\text{seg}}} \right\rfloor \right\},$$

$$P' = \min\{0.99, [1 - (1 - P)^{L_{\text{seg}} \times 8}]\}.$$

The maximum number of retransmissions allowed for the CP segment can be formulated as

$$N_{\text{max\_CP\_retran}} = \max\{3, \lceil \log_{\min\{0.99, [1-(1-P)^{L_{\text{seg}} \times 8}]\}}(10^{-6}) \rceil\}, \tag{11}$$

$$N_{\text{max\_RS\_retran}} = \max\{3, \lceil \log_{[1-(1-P)^{L_{\text{RS}} \times 8}]}(10^{-6}) \rceil\}. \tag{12}$$

Once any of the above maximum number of retransmissions is exceeded, the current session transmission is canceled. This means that the number of block transmissions has increased by one, which increases the file delivery delay.

Algorithm 1 Modeling red data transmission based on LTP reliable mechanism

$N_{\text{loss\_pre}} = N_{\text{seg\_tr}}$; //The number of segments to be sent in this round (i.e., $N_{\text{seg\_tr}}$) is temporarily stored in $N_{\text{loss\_pre}}$ for use by Algorithm 5.
$T_{\text{trans\_file}} = T_{\text{trans\_file}} + N_{\text{seg\_tr}} \times L_{\text{link\_pack}}/R_{\text{est\_red}}$; //Add the transmission delay of current red data to $T_{\text{trans\_file}}$.
$N_{\text{RS\_block}} = N_{\text{RS\_block}} + C$; //Increase the number of report segments for the current block by C, which indicates the number of RS excluding retransmission in this round.
if $N_{\text{RS\_block}} > N_{\text{max\_RS\_num}}$ **then**
  $N_{\text{block}} = N_{\text{block}} + 1$; //Increase the number of blocks which will to be transferred.
else
  $N_{\text{loss}} = 0$; //N_{loss} records the number of segments lost in this round.
  flag = 0; //flag is used to record whether the first CP of each block is lost, and 1 means lost.
  for $m = 1$ to $N_{\text{seg\_tr}}$ **do**
    if rand $< P_{\text{seg}}$ **then**
      //It means that the segment is lost.
      $N_{\text{loss}} = N_{\text{loss}} + 1$;
      if $m == N_{\text{seg}}$ **then**
        //Indicate that the last segment is lost in the first round.
        flag = 1;
      end
    end
  end
end

2.1 Overall framework of the model

In this study, we use the roulette wheel method instead of statistical averaging to model the interaction process of LTP reliable transmission, which can better reflect the random characteristics in the transmission process. In Figure 2, the file delivery delay model of LTP reliable transmission is illustrated using a flow chart. The inputs required for this model include bundle size, block limit size, BER, and file size. The iteration termination condition in this model is determined by setting a maximum number of iterations. In our experiment, the maximum value is set to 16 for obtaining each numerical result [36]. Once the iteration termination condition is met, the algorithm concludes and produces the file delivery delay as output. This delivery delay is determined by summing up all iteration results and dividing it by the maximum number of iterations. In each iteration, each block is delivered sequentially, as there is only one session in this model to simplify the process. Therefore, the current block cannot be transmitted until all preceding blocks have been successfully delivered. The current iteration continues as long as there is at least one undelivered block, indicated by $N_{\text{block}} \geqslant 1$. The number of transmissions of every RS is recorded in N_{RS} with the initial value of 0. $N_{\text{round\_block}}$ represents the number of times the current block has been transmitted, with an initial value of 0. $N_{\text{seg\_tr}}$ is used to record the number of segments to be sent in the current round and its initial value is N_{seg}. A sending process is called a round, that is, the process from sending data to receiving a report segment from the receiver. As long as $N_{\text{seg\_tr}}$ is greater than or equal to 1, that is, there are still segments in the current block that have not been delivered correctly, the following loop process is executed. The processing mainly includes three functional modules, namely the red data transmission, control information exchange (followed by CP, RS, and RA control information processing), and the number of retransmitted segments in the next round. For every block, if there are segments that have not been delivered correctly, the next round needs to be executed unless the maximum number of retransmissions is exceeded. Each functional module is described in details in this section.

2.2 LTP red data transmission model

As mentioned, the model uses a roulette wheel to simulate random packet loss or damage in spatial transmission channels, which can better reflect the random characteristics of channel packet loss compared to statistical methods. Red data transmission processing is proposed and described in Algorithm 1.

As depicted in Algorithm 1, the counter for lost segments increases by 1 if the generated random number is less than P_{seg} for each segment of the current block during transmission. In each block, the last segment serves as a CP during the first round. However, if the last segment is lost during transmission, it is necessary to record the event. This record helps determine the appropriate actions and adjustments needed during the subsequent rounds. In the delivery process of each block, besides the CP for the first round, the subsequent CP in the data retransmission rounds is formed by selecting a few bytes from the last segment. The CP processing model is described in Algorithm 2. The RS counter of the current block is increased in each round, and it is necessary to determine whether the RS counter exceeds the

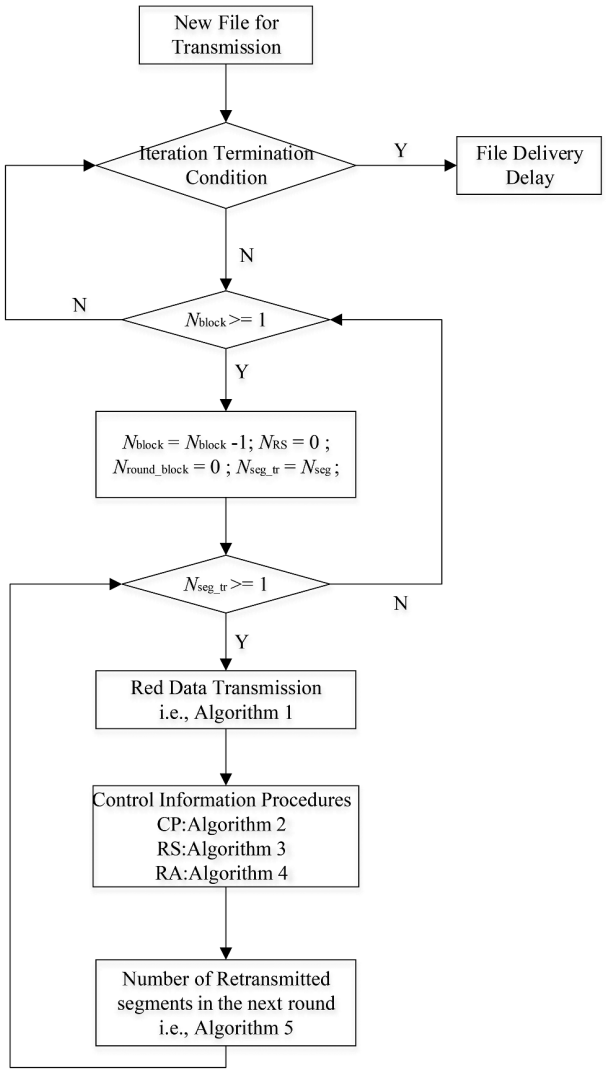

Figure 2 Flow chart of data delivery delay model for LTP reliable transmission.

maximum transmission limit before the next retransmission. If the RS counter exceeds the maximum transmission limit, the current session is canceled, and the number of blocks that need to be sent again is incremented by 1. To simplify the model, only one session is used in this algorithm. The transmission of the current block continues until all segments are successfully delivered within the maximum round limit. Once all the segments are successfully transmitted, the transmission of the current block is completed, and the transmission of the next block can commence. It is worth noting that if the RA responded to the previous RS is lost, and the previous RS is still retransmitted, it will not only cause the current RS to wait for transmission, but also cause the CP to be retransmitted.

2.3 Control information interaction model

According to the reliable transmission mechanism of LTP, the retransmission timer is activated after the CP is sent in each round. If the corresponding RS is not received by the sender within the retransmission timer interval, the CP needs to be retransmitted. As a result, the timer interval is accumulated into a file delivery delay. Let T_{RTO} be the waiting interval for the retransmission timeout (RTO), and $T_{\mathrm{RA\_add}}$ represent the additional delay added during the processing of control information RA. It is important to mention that the retransmission of CP is not indefinite or endless. The current session is canceled if the maximum retransmission limit, denoted as $N_{\mathrm{max\_CP\_retran}}$, is exceeded. Algorithm 2 presents the modeling of CP processing, which primarily consists of two stages. One stage of CP processing is the CP of the first round, which refers to the first CP for each block. The other stage is the CP in subsequent

Algorithm 2 Modeling of control information CP processing

```
if N_seg_tr == N_seg then
    //Indicate that this is the first round of the current block.
    if flag == 1 then
        //The CP of the first round is lost.
        T_prop_file = T_prop_file + T_RTO + T_RA_add;
        while rand < P_seg do
            //Retransmit the first CP.
            N_CP = N_CP + 1; //N_CP records the times of CP transmissions.
            if N_CP − 1 > N_max_CP_retran then
                //Exceed the maximum retransmission limit of CP.
                N_block = N_block + 1;
                break;
            else
                T_prop_file = T_prop_file + T_RTO;
            end
        end
        T_prop_file = T_prop_file + T_RTT;
    else
        T_prop_file = T_prop_file + T_RTT + T_RA_add;
    end
else
    //The subsequent rounds of each block except the first round.
    while rand < P_CP do
        //CP loss.
        N_CP = N_CP + 1;
        if N_CP − 1 > N_max_CP_retran then
            N_block = N_block + 1;
            break;
        else
            T_prop_file = T_prop_file + T_RTO;
        end
    end
    T_prop_file = T_prop_file + T_RTT + T_RA_add;
end
```

Algorithm 3 Modeling of control information RS processing

```
N_RS = 0;
while rand < P_RS do
    //RS loss.
    N_RS = N_RS + 1; //N_RS records the number of RS transmissions.
    N_CP = N_CP + 1; //The loss of RS also results in CP retransmission.
    T_prop_file = T_prop_file + T_RTO;
    if N_seg_tr == N_seg & N_RS − 1 > N_max_RS_retran then
        //The first round only considers the number of RS retransmissions.
        N_block = N_block + 1;
        break;
    else
        if N_RS − 1 > N_max_RS_retran or N_CP − 1 > N_max_CP_retran then
            //In addition to the first round, the maximum retransmission limits of both RS and CP should be considered in
            subsequent rounds.
            N_block = N_block + 1;
            break;
        end
    end
end
```

rounds, excluding the first round. The main difference between the two situations is the size of the CP. The delivery delay increases by T_{RTO} when CP is lost, otherwise T_{RTT} is accumulated. In Algorithm 2, the effect of RA loss on CP is considered.

Similarly, in Algorithm 3, RS also has a corresponding retransmission timer that is activated after RS is sent. If the confirmation message RA is not received within the timer interval, it becomes necessary to resend RS, leading to the same amount of increase to the delivery delay as the CP processing. There is also a limit on the maximum number of RS retransmissions, i.e., $N_{\mathrm{max\_RS\_retran}}$. The corresponding session is canceled if the limit is exceeded. The current CP is also retransmitted when RS is retransmitted since there is no corresponding RS response. It is also subject to the maximum number of retransmissions in RS processing. It is important to note that the corresponding CP retransmission is not limited when the first RS is lost in the first round for each block, as shown in Algorithm 3.

In this model, the last control message involved is the RA, as indicated in Algorithm 4. In each session,

Algorithm 4 Modeling of control information RA processing

```
k = 0; //Indicate the sending times of the same RA.
T_RA_add = 0; //Record the additional delay caused by the loss of RA.
while rand < P_RA do
    //RA loss.
    k = k +1;
    N_RS = N_RS +1; //The loss of the current RA causes the retransmission of the previous RS.
    T_RA_add = T_RA_add + T_RTO;
    if N_RS − 1 > N_max_RS_retran then
        N_block = N_block +1;
        break;
    end
end
```

Algorithm 5 The number of segments to be transmitted in the next round

```
if flag == 1 then
    //The CP of the first round is lost.
    N_seg_tr = N_seg − 1; //Retransmission is required except the CP itself in this case.
else
    //The CP of the current block is delivered correctly in the first round.
    if N_RS == 1 then
        //RS does not need retransmission in the current round.
        N_seg_tr = N_loss; //The number of segments to be transmitted in the next round is the amounts of lost segments in the
        current round.
    else
        //RS is retransmitted in the current round.
        if N_seg_tr == N_seg then
            //The first round of the current block.
            N_seg_tr = N_loss;
        else
            //The subsequent rounds except the first round for the current block.
            N_seg_tr = N_loss_pre; //The number of transmission segments in the next round is the same as this time.
        end
    end
end
```

if the current RA is lost, the previous RS is retransmitted, resulting in the waiting of the current RS before it can be sent. Therefore, if the RA is lost, it may lead to an increase in total delivery delay as one or more RS retransmission timer intervals are extended. This is influenced by the number of retransmissions of previous RS due to failed RA, and the impact of RA is accumulated onto $T_{\text{RA\_add}}$. Additionally, this process is also constrained by the RS retransmission threshold.

In addition, it is also important to estimate the number of segments to be retransmitted in the next round, which is one of the key factors affecting overall delivery delay. Algorithm 5 presents the solution model. During the transmission process of each block, if the CP is lost in the first round, in the subsequent round of CP retransmission, the RS only confirms the CP itself and does not confirm the data segments received in the first round. In this case, all the data segments except CP need to be retransmitted in the subsequent round, i.e., $N_{\text{seg}} - 1$. Otherwise, the number of segments to be transmitted in the next round is equal to the number of lost red segments in the current round, regardless of the number of times the corresponding RS has been transmitted. In the subsequent rounds, the number of segments to be retransmitted in the next round is determined according to the RS retransmission times of the current round. If RS is successfully delivered for the first time (i.e., without retransmission), the segments to be retransmitted in the next round are the lost segments in the current round. Otherwise, it should be the segments to be transferred in the current round.

3 Model validation and performance analysis

In this section, we provide numerical values and experimental results of BP/LTP to validate the model for reliable file delivery over space communication channels. The validation experiments were conducted on a PC-based networking test-bed [21, Figure 1], which has been validated in prior work [21, 22, 27, 28, 37].

The proposed file transfer experiment utilizes the protocol and parameter settings outlined in Table 2. The interplanetary overlay network (ION) distribution v4.1.1 [38] developed by NASA's JPL, California Institute of Technology, included the implementations of the BP and LTP protocols. ION, which is a software implementation of the DTN protocol suite, targets space internetworking and deep-space

Table 2 Model parameters and protocol settings for numerical analysis

Parameter	Setting
Protocol stack adopted	BP/LTP (custody option disabled)
LTP red/green setting	100% red data for reliable delivery
File size	1 MB
Bundle size	1 KB, 2 KB, 4 KB, 5 KB, 10 KB, 20 KB, 40 KB, 50 KB, 100 KB, 200 KB, 500 KB, 1 MB
block aggregation size threshold	2 KB, 4 KB, 5 KB, 10 KB, 20 KB, 40 KB, 50 KB, 100 KB, 200 KB, 500 KB, 1 MB
Segment size	200 Byte, 400 Byte, 600 Byte, 800 Byte, 1000 Byte, 1200 Byte, 1400 Byte
Number of sessions	1
BER	10^{-7}, 10^{-6}, 10^{-5}
Data channel rate	10 Mbit/s
ACK channel rate	100 kbit/s
RTT	2.56 s
RTO	3 s

Table 3 (Color online) Data payload length actually encapsulated in LTP output block

Bundle size (Byte)	Block aggregation size threshold (Byte)										
	2K	4K	5K	10K	20K	40K	50K	100K	200K	500K	1M
1K	2K	4K	5K	10K	20K	40K	50K	100K	200K	500K	1M
2K	2K	4K	6K	10K	20K	40K	50K	100K	200K	500K	1M
4K	4K	4K	8K	12K	20K	40K	52K	100K	200K	500K	1M
5K	5K	5K	5K	10K	20K	40K	50K	100K	200K	500K	1M
10K	10K	10K	10K	10K	20K	40K	50K	100K	200K	500K	1M
20K	20K	20K	20K	20K	20K	40K	60K	100K	200K	500K	1M
40K	40K	40K	40K	40K	40K	40K	80K	120K	200K	520K	1M
50K	50K	50K	50K	50K	50K	50K	50K	100K	200K	500K	1M
100K	100K	100K	100K	100K	100K	100K	100K	100K	200K	500K	1M
200K	200K	200K	200K	200K	200K	200K	200K	200K	200K	600K	1M
500K	500K	500K	500K	500K	500K	500K	500K	500K	500K	500K	1M
1M	1M	1M	1M	1M	1M	1M	1M	1M	1M	1M	1M

communications. In our experiment, we configured a one-way link delay of approximately 1.28 s to simulate the unavoidable propagation delay in cislunar communication scenarios. The three channel BERs, 10^{-7}, 10^{-6}, 10^{-5}, generated using a typical AWGN model, were adopted for our experiments.

3.1 Verification of the data payload length actually encapsulated in LTP output block

The payload length in this paper refers to the length of the data part encapsulated in the LTP output block excluding the overhead of the BP layer, which depends on the bundle size and block aggregation size threshold. The length of the data load encapsulated in each block determines the size of the sending window, which directly affects the end-to-end delivery delay and transmission efficiency. In our model, Eqs. (4)–(6) describe the relationship between the actual output block length and the block aggregation threshold and bundle size. The outcomes obtained through ION and Wireshark packet capture analysis are presented in Table 3.

In Table 3, the bundle size does not include the overhead of BP. It is observed that when the bundle size exceeds the block aggregation size threshold, the length of the data payload encapsulated in the block is determined by the bundle size. In such a case, the length of the payload is equal to the bundle size, as shown in green in Table 3. When the bundle size is smaller than the block aggregation size, the length of the data payload encapsulated in the block depends on the block aggregation size threshold. Only when the block aggregation size threshold is an integer multiple of bundle size, the data payload length encapsulated in the block is equal to the block aggregation size threshold. Otherwise, the data payload length is the smallest value within the set that is greater than the block aggregation size threshold and an integer multiple of the bundle size, as shown in red in Table 3. For instance, when the bundle size is 2 KB and the block aggregation size threshold is 5 KB, the actual length of the data payload encapsulated within the LTP block is 6 KB, i.e., a block contains three bundles as shown in Figure 3. This verifies the correctness of (4)–(6). The results presented in Table 3 also demonstrate that the bundle is not

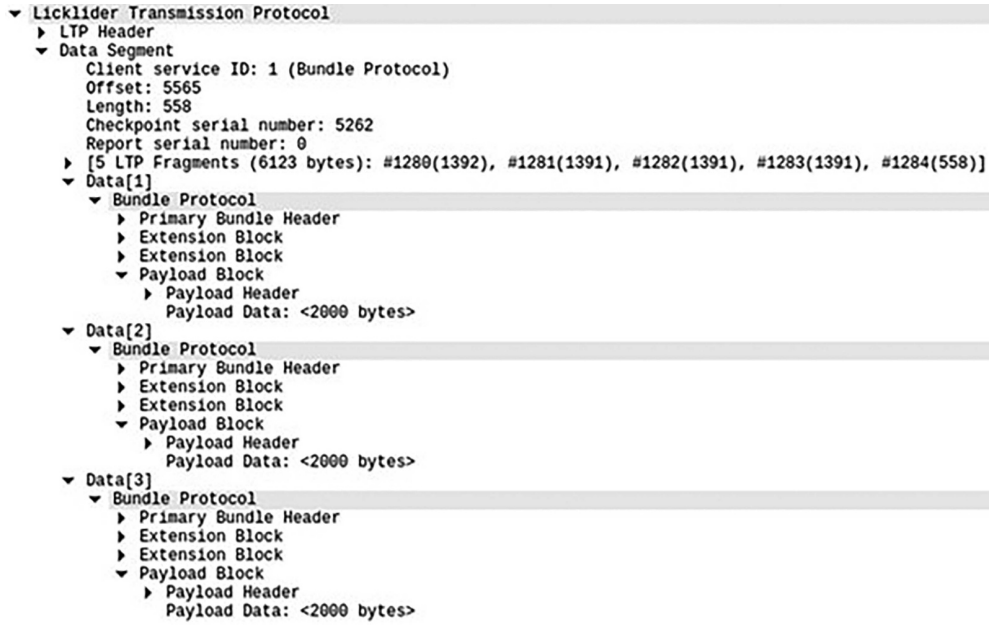

Figure 3 The number of bundles contained in a block (with bundle size of 2 KB and block aggregation size threshold of 5 KB)

fragmented or split. The length of the data payload in the actual output block is an integer multiple of the bundle load.

3.2 Influence of bundle size on file delivery delay

Figure 4 presents a comparison of file delivery delay using BP/LTP between model's numerical values and experimental results, and the bundle size is shown in Table 2. The considered block aggregation sizes are 5 KB, 50 KB, 500 KB, and 1 MB with varying data channel error rates of 10^{-7}, 10^{-6}, 10^{-5}, and the segment size is 1400 Byte. Figure 4 shows the relationship between file delivery latency and packet size. The numerical values are close to the experimental results. The file delivery delay rises as the channel BER increases. Obviously, a higher BER during transmission results in more blocks being corrupted or lost, which leads to more retransmission events required to successfully deliver a single block and therefore, an entire file. The additional transmission rounds caused by increased retransmission events contribute to extra time that is included in the file delivery delay. This leads to a much longer file delivery delay at BERs of 10^{-5} than of 10^{-7} and 10^{-6}. In comparison, the delivery delay is smaller when the block aggregation size is set to 1 MB in Figure 4. It is observed that the delivery delay experiences a notable decrease as the bundle size increases for block aggregation size of 5 KB, 50 KB, and 500 KB, as shown in Figures 4(a)–(c). Especially, in Figures 4(a) and (b), there is an exponential decrease, and the smallest delivery delay is attained when the bundle size is 1 MB. However, this is different for block aggregation sizes of 1 MB. As shown in Figure 4(d), the file delivery delay does not continue to decrease with an increased bundle size, and there is no noteworthy difference in delivery delay for the same BER. Figure 4 also demonstrates that there is not a significant difference in delivery delay for each BER when the bundle size is less than or equal to the block aggregation size. Because estimated output block size (i.e., $\overline{L_{\text{est\_exp\_block}}}$) is not affected by bundle size and is equal to the expected output block size, which corresponds to the block aggregation size threshold $L_{\text{agg\_size\_limit}}$, as stated in (6). But the block size is equal to the bundle size in the cases of bundles greater 5 KB, 50 KB, 500 KB, and 1 MB in Figure 4. The results are completely consistent with the model analysis, that is, Eq. (4) in Section 2. In Figures 4(b)–(d), it is evident that the delivery delay curve fluctuates in an oscillating manner as the bundle size varies, particularly when the bundle size is below a certain threshold value. This phenomenon also exists in Figure 4(a); it cannot be visually observed due to its dense distribution. This is because the block size is influenced by the bundle size and block aggregation size threshold, as shown in (4). For example, in Figure 4(b), when the bundle size is 40 and 50 KB, the block size is 80 and 50 KB, respectively, according to (4), and it is consistent with the block size captured on our experimental testing platform. Therefore, the delivery delay is shorter for bundle size of 40 KB compared to bundle sizes of 50 KB.

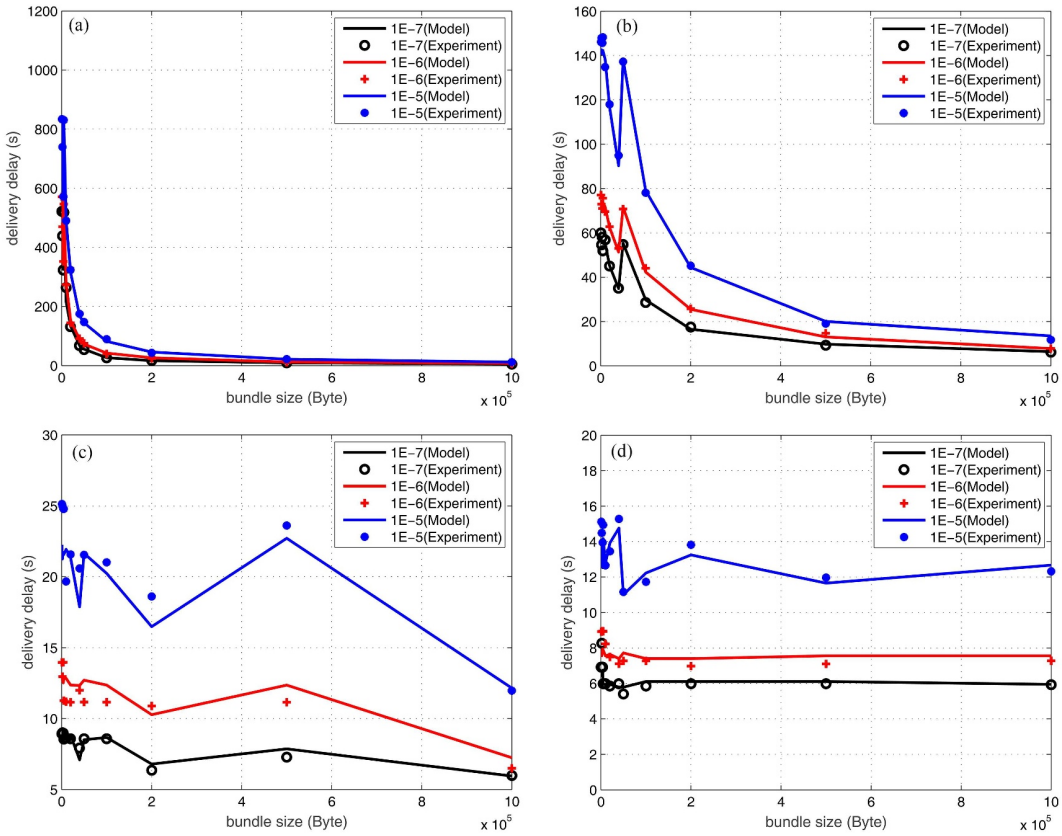

Figure 4 (Color online) Impact of bundle size on delivery delay. Block aggregation size threshold of (a) 5 KB, (b) 50 KB, (c) 500 KB, and (d) 1 MB.

As observed, the bundle size plays a crucial role in transmission performance, especially when the block aggregation size threshold is small. The overall trend of file delivery delay decreases as the bundle size exceeds the block aggregation size. It implies that a larger bundle size offers greater benefits in improving transmission performance when the block aggregation size threshold is small. In contrast, for the larger block aggregation size threshold, the bundle size has little impact on transmission performance.

3.3 Impact of block aggregation size threshold on file delivery delay

Figure 5 displays the predicted file delivery delay of LTP for successfully delivering a 1 MB file under varying data channel error rates (i.e., the BERs of 10^{-7}, 10^{-6}, and 10^{-5}), considering different block aggregation sizes. In Figure 5, the reason of the delivery delay fluctuation within the range of small block aggregation size is the same as in Figure 4 and is not repeated here. In Figures 5(a)–(c), it can be observed that the overall trend of delivery delay decreases with the increase of block aggregation size. This implies that larger aggregation block sizes result in better transmission performance. In Figures 5(a) and (b), it is evident that the transmission performance shows an exponential increase, and larger block sizes result in higher efficiency. However, the file delivery delay will not continue to decrease as the block aggregation size threshold increases, as shown in Figure 5(d).

In Figure 5, it is obvious that when the block aggregation size threshold is less than the bundle size, the file delivery delay is almost the same. On the contrary, when the block aggregate size threshold is greater than the bundle size, as the block aggregation size threshold increases, the LTP sending window becomes larger, improving the transmission efficiency, which is evident in Figures 5(a)–(c). It is worth noting that increasing the block size beyond 1 MB does not improve transmission performance. Therefore, the upper limit of the block aggregation size threshold is determined by the file size. On the other hand, the minimum block size is set to the bundle size, meaning that the bundle remains unsegmented during the encapsulation block processing.

Tables 4–6 present a summary of the approximate rounds required for successful delivery of the entire file in our experiment, along with the corresponding BERs of 10^{-7}, 10^{-6}, and 10^{-5}, respectively. The

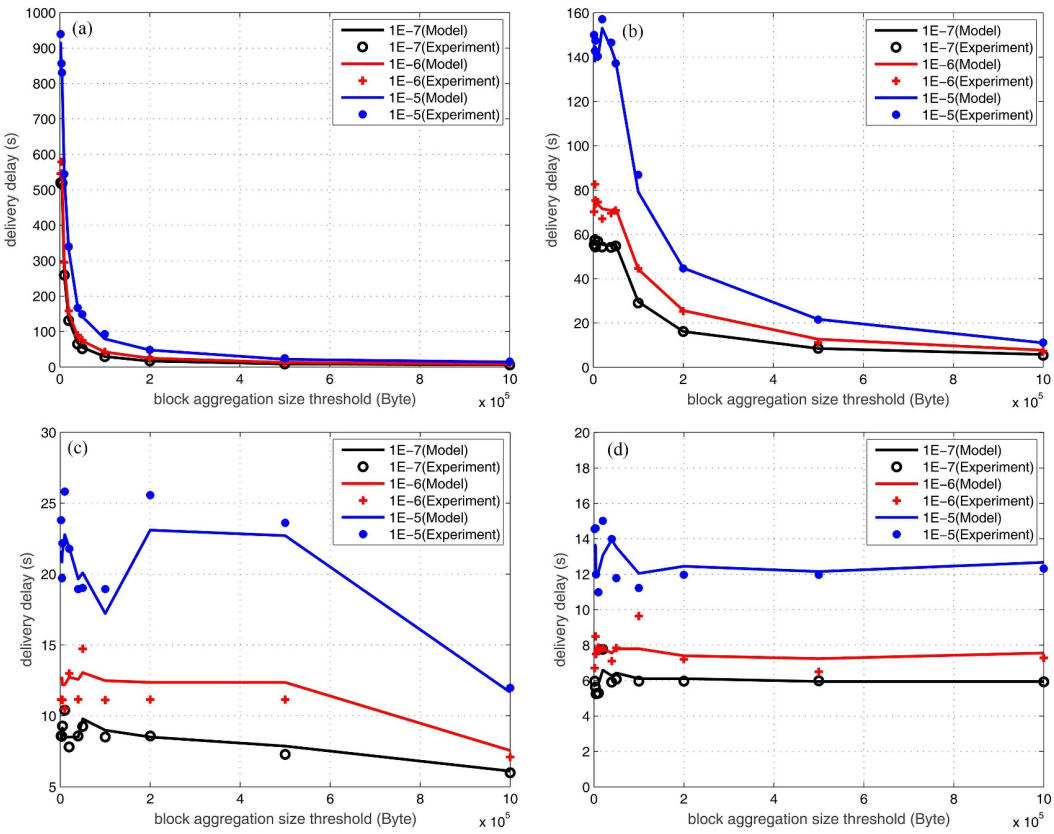

Figure 5 (Color online) Impact of block aggregation size threshold on delivery delay. Bundle sizes of (a) 5 KB, (b) 50 KB, (c) 500 KB, and (d) 1 MB.

Table 4 (Color online) An approximate number of rounds required for successful delivery under BER of 10^{-7}

Bundle size (Byte)	Block aggregation size threshold (Byte)										
	2K	4K	5K	10K	20K	40K	50K	100K	200K	500K	1M
1K	512.7	255.1	203.5	102.8	52.8	26.6	23.5	11.7	7.8	3.5	2.7
2K	502.9	251.2	171.4	101.1	51.3	26.3	21.4	11.7	6.6	3.5	3.2
4K	254.3	252.1	126.5	86.8	50.3	27.3	22.7	11.4	6.3	3.5	2.7
5K	203.0	202.0	202.5	101.4	51.3	25.3	20.3	11.4	6.6	3.3	2.3
10K	101.6	101.5	103.2	100.4	51.3	26.1	22.2	12.4	6.3	3.4	2.3
20K	51.5	50.5	51.3	51.3	52.3	25.2	17.6	11.4	6.4	3.4	2.3
40K	26.9	28.4	26.2	26.2	25.2	26.1	13.7	10.6	6.4	3.1	2.3
50K	21.6	22.4	21.2	22.2	21.2	21.2	21.4	11.4	6.3	3.4	2.1
100K	11.4	10.4	10.2	12.1	12.0	12.1	11.2	11.4	6.4	3.4	2.3
200K	6.3	6.4	6.6	6.6	6.5	6.4	6.9	6.4	6.4	2.5	2.3
500K	3.3	3.3	3.6	4.1	3.1	3.3	3.6	3.3	3.4	2.8	2.3
1M	2.3	2.2	2.1	2.1	3.0	2.3	2.4	2.3	2.3	2.3	2.3

approximate number of rounds required can be calculated by dividing the experimental delivery delay by RTT. Tables 4–6 visually demonstrate that increasing the size of the bundle or block aggregation threshold leads to a reduction in the required number of rounds, which is supported by the results shown in green in Tables 4–6 within the acceptable range. However, when the sizes of block and bundle are unreasonably small, the number of required rounds increases by hundreds of times. It is also obvious that as the bit error rate increases, the number of rounds required also increases under the same conditions.

3.4 Impact of segment size for file delivery delay

Figure 6 shows the results of model analysis and experimental platform under three transmission channels with variable segment sizes. It is worth noting that the segment size should not exceed the MTU

Table 5 (Color online) An approximate number of rounds required for successful delivery under BER of 10^{-6}

Bundle size (Byte)	Block aggregation size threshold (Byte)										
	2K	4K	5K	10K	20K	40K	50K	100K	200K	500K	1M
1K	529.4	267.8	223.1	116.9	60.6	35.2	30.1	17.8	10.8	5.5	3.5
2K	522.8	269.4	183.5	110.8	58.4	36.6	28.5	17.6	10.5	5.1	3.5
4K	272.4	268.8	137.6	100.1	58.6	33.9	27.8	17.5	10.4	5.5	3.5
5K	213.0	226.1	213.4	115.3	61.8	34.9	29.5	16.7	9.9	4.4	3.5
10K	110.9	116.4	107.0	108.5	58.5	34.5	27.2	15.9	9.4	4.4	3.2
20K	57.5	61.7	56.4	62.8	61.6	33.7	24.5	15.4	10.0	4.4	2.9
40K	33.5	37.2	35.6	33.2	32.8	35.2	20.7	17.1	9.7	4.7	2.8
50K	27.4	32.3	29.4	29.1	26.2	27.2	27.7	17.4	9.9	4.4	2.8
100K	15.4	16.4	15.8	13.0	16.2	15.1	17.2	15.4	9.4	4.4	2.8
200K	11.7	10.4	9.4	10.1	11.1	9.4	10.1	11.4	10.3	4.3	2.7
500K	4.4	4.3	4.3	4.1	5.1	4.4	5.8	4.3	4.4	4.4	2.8
1M	2.6	3.3	2.9	3.1	3.1	2.8	3.1	3.8	2.8	2.5	2.8

Table 6 (Color online) An approximate number of rounds required for successful delivery under BER of 10^{-5}

Bundle size (Byte)	Block aggregation size threshold (Byte)										
	2K	4K	5K	10K	20K	40K	50K	100K	200K	500K	1M
1K	746.4	419.0	325.7	221.8	133.1	35.0	57.1	37.8	21.2	9.8	5.9
2K	739.6	405.6	288.9	213.3	125.2	69.0	57.8	37.5	20.7	9.8	5.7
4K	398.6	390.9	223.3	184.9	140.8	66.2	56.9	31.8	19.9	9.7	5.4
5K	366.9	334.5	324.4	212.5	132.6	65.2	57.9	36.2	18.8	9.7	5.8
10K	214.9	210.8	191.4	200.2	120.1	70.6	52.7	29.2	17.5	7.7	4.9
20K	124.4	125.9	126.4	118.2	117.7	66.1	46.1	31.3	17.8	8.4	5.3
40K	83.4	80.2	68.3	67.3	74.3	66.3	37.1	30.3	20.4	8.0	6.0
50K	58.6	55.8	57.6	54.8	61.4	57.3	53.6	17.4	17.4	8.4	4.4
100K	33.4	32.5	34.9	31.5	32.6	31.2	30.5	15.4	19.0	8.2	4.6
200K	18.2	20.2	16.8	18.2	18.6	18.6	17.7	11.4	17.5	7.3	5.4
500K	9.3	7.7	8.7	10.1	8.5	7.4	7.4	4.3	10.0	9.2	4.7
1M	5.7	5.7	4.7	4.3	5.9	5.5	4.6	3.8	4.7	4.7	4.8

(1400 Byte in this paper) in the link layer, otherwise the fragmentation and reorganization of the link layer cause an additional processing delay. Therefore, the upper limit for the segment size is dictated by the MTU in the link layer. The bundle size is 5 KB, 50 KB, 500 KB, and 1 MB, respectively, with block size equal to the bundle size. As depicted in Figure 6, an increase in the size of bundle and block leads to a reduction in file delivery delay. Compared to Figures 6(a) and (b), the file delivery delay in Figures 6(c) and (d) is approximately one order of magnitude lower. It is evident that the larger segment has a higher likelihood of enhancing transmission performance when the channel quality is good, such as 10^{-7} and 10^{-6}. In this case, setting the segment size to 1400 Byte is considered the most optimal choice, because the larger segment results in a smaller total header overhead, leading to higher transmission efficiency. However, in the case where the channel quality is poor, such as BER of 10^{-5}, the situation is different. In other words, the larger segment size may not always achieve good transmission performance, which will be more obvious for the transmission of large-sized files.

As shown in Figure 6, for the channel with BER of 10^{-5}, the delivery delay of files typically exhibits an upward trend with an increase in segment size in Figures 6(a) and (b). However, in Figures 6(c) and (d), the delivery delay generally shows a downward or stable trend with an increase in segment size. The minimum file delivery delay with BER of 10^{-5} is obtained at segment sizes of 400 Byte, 200 Byte, 800 Byte, 1 KB, respectively, and corresponding minimum values are indicated by a circle in Figure 6. The choice of segment size, whether set to the optimal value or a larger one (in this case, the maximum segment size is 1400 Byte), has a significant impact on file delivery delay as shown in Figures 6(a) and (b). However, in Figures 6(c) and (d), the difference is not obvious. This is because in a poor channel, the larger bundle or block has a greater impact on file transmission than segment size. Therefore, the segment size should be set to the maximum value when the channel error rate is low (i.e., BERs of 10^{-5} and 10^{-6}), while for higher error rates, for example 10^{-5}, the selection of segment size should also consider the size of blocks or bundles. It is true that larger segment size can be advantageous for larger

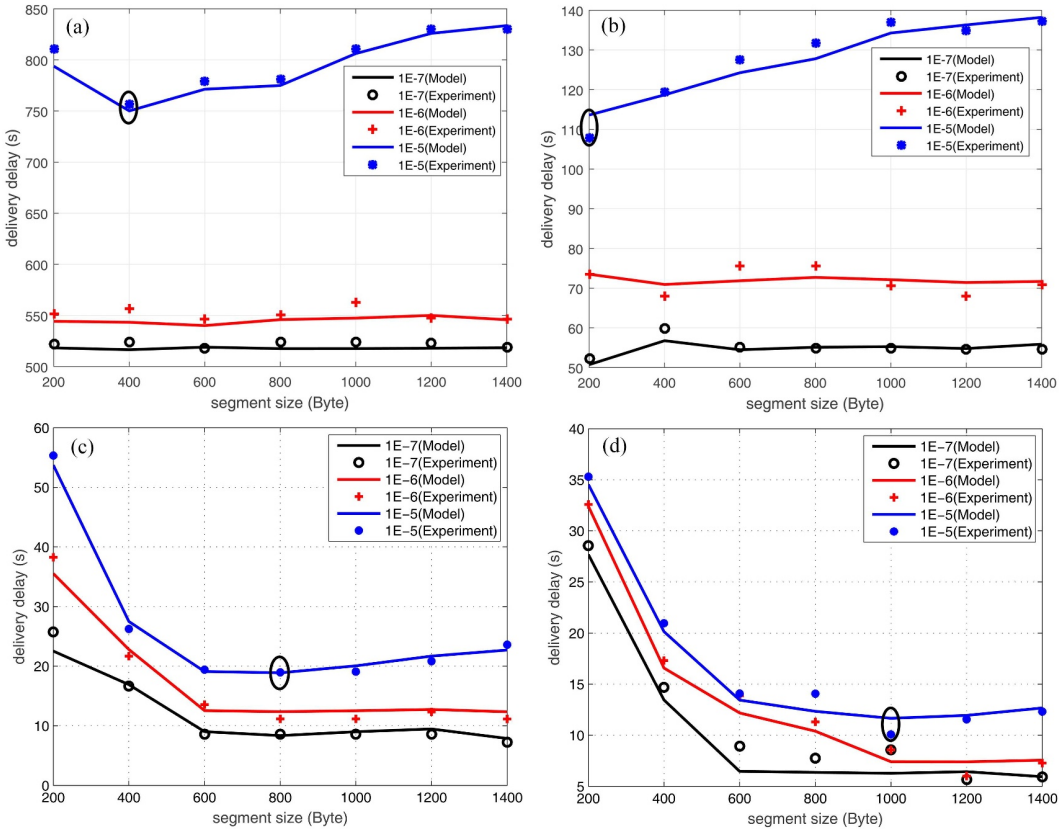

Figure 6 (Color online) Impact of segment size on delivery delay. Bundle size and block aggregation size threshold of (a) 5 KB, (b) 50 KB, (c) 500 KB, and (d) 1 MB.

bundles or blocks, while smaller segment size is more suitable for smaller bundles or blocks to improve transmission performance in poor channels.

4 Summary and conclusion

This paper presents a BP/LTP delivery delay model for reliable transmission over space network channels. The study also analyzes the impact of cross layer packet size on LTP reliable transmission. The key cross-layer packets include bundle, block, and segment, and they all have an important impact on the protocol transmission performance. However, the study results indicate that compared to segment size, bundle size and block size have a more pronounced influence on the transmission performance of LTP.

The block size of the LTP layer depends on bundle size and block aggregation size threshold. The study indicates that the impact of bundle size and block aggregation size on delivery delay is similar—as long as one of them is set larger, the delivery delay is optimized. This is because a larger output block size has the same effect to the transmission performance as a larger sending window size of LTP. On the premise that ($L_{\text{agg\_size\_limit}} \geqslant R_{\text{est\_red}} \times T_{\text{agg\_time\_limit}}$) is satisfied, when the bundle size is less than or equal to the block aggregation size threshold, the performance difference with respect to variations of bundle size is not significant. On the contrary, increasing the bundle size in aggregation time $T_{\text{agg\_time\_limit}}$ is beneficial for reducing delivery delay, which is correct for the block aggregation size threshold setting. For transmission over a less lossy channel, it is more appropriate to select a segment size corresponding to the MTU of the link layer. When the channel is lossy, it is optimal to configure the segment size to be smaller than the MTU. As a part of the future work, we will take random link interruption into consideration in our subsequent modeling. With that, the impact of random link interruption on reliable transmission of LTP in space networks will be analyzed.

Acknowledgements This work was supported by Chongqing Key Laboratory of Mobile Communications Technology (Grant No. cqupt-mct-202203).

References

1 Abdelsadek M Y, Chaudhry A U, Darwish T, et al. Future space networks: toward the next giant leap for humankind. IEEE Trans Commun, 2023, 71: 949–1007
2 Alhilal A, Braud T, Hui P. The sky is NOT the limit anymore: future architecture of the interplanetary internet. IEEE Aerosp Electron Syst Mag, 2019, 34: 22–32
3 Choudhari C, Niture D. Disruption tolerant network (DTN) for space communication: an overview. In: Proceedings of IEEE 7th International Conference for Convergence in Technology (I2CT), Mumbai, 2022
4 Ha T, Lee D, Oh J, et al. DTN-based multi-link bundle protocol architecture for deep space communications. In: Proceedings of the 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, 2022. 1164–1166
5 Yang L, Liang J, Wang R, et al. A study of Licklider transmission protocol in deep-space communications in presence of link disruptions. IEEE Trans Aerosp Electron Syst, 2023, 59: 6179–6191
6 Alessi N, Caini C, de Cola T, et al. Packet layer erasure coding in interplanetary links: the LTP erasure coding link service adapter. IEEE Trans Aerosp Electron Syst, 2020, 56: 403–414
7 Yang L, Wang R, Zhou Y, et al. An analytical framework for disruption of Licklider transmission protocol in Mars communications. IEEE Trans Veh Technol, 2022, 71: 5430–5444
8 Guidotti A, Vanelli-Coralli A, Conti M, et al. Architectures and key technical challenges for 5G systems incorporating satellites. IEEE Trans Veh Technol, 2019, 68: 2624–2639
9 Burleigh S C, Hooke A, Torgerson L, et al. Delay-tolerant networking: an approach to interplanetary Internet. IEEE Commun Mag, 2003, 41: 128–136
10 Scott K, Burleigh S C. Bundle protocol specification. IETF Request for Comments RFC 5050, 2007. doi: 10.17487/RFC5050
11 Burleigh S C, Ramadas M, Farrell S. Licklider transmission protocol-motivation. IRTF Internet Draft, 2007. doi: 10.17487/RFC5325
12 Ramadas M, Burleigh S C, Farrell S. Licklider transmission protocol specification. Internet RFC 5326, 2008. doi: 10.17487/RFC5326
13 Consultative Committee for Space Data Systems. Licklider Transmission Protocol (LTP) for CCSDS. CCSDS 734.1-B-1, 2015. https://public.ccsds.org/Pubs/ 734x1b1.pdf
14 Wu S H, Li D Q, Jiao J, et al. CS-LTP-Spinal: a cross-layer optimized rate-adaptive image transmission system for deep-space exploration. Sci China Inf Sci, 2022, 65: 112303
15 Wang R, Wei Z, Zhang Q, et al. LTP aggregation of DTN bundles in space communications. IEEE Trans Aerosp Electron Syst, 2013, 49: 1677–1691
16 Sabbagh A, Wang R, Zhao K, et al. Bundle protocol over highly asymmetric deep-space channels. IEEE Trans Wireless Commun, 2017, 16: 2478–2489
17 Wang R, Qiu M, Zhao K, et al. Optimal RTO timer for best transmission efficiency of DTN protocol in deep-space vehicle communications. IEEE Trans Veh Technol, 2017, 66: 2536–2550
18 Zhao K, Wang R, Burleigh S C, et al. Performance of bundle protocol for deep-space communications. IEEE Trans Aerosp Electron Syst, 2016, 52: 2347–2361
19 Yu Q, Sun X, Wang R H, et al. The effect of DTN custody transfer in deep-space communications. IEEE Wireless Commun, 2013, 20: 169–176
20 Yang G, Wang R, Sabbagh A, et al. Modeling optimal retransmission timeout interval for bundle protocol. IEEE Trans Aerosp Electron Syst, 2018, 54: 2493–2508
21 Wang R, Burleigh S C, Parikh P, et al. Licklider transmission protocol (LTP)-based DTN for cislunar communications. IEEE ACM Trans Networking, 2011, 19: 359–368
22 Sun X, Yu Q, Wang R, et al. Performance of DTN protocols in space communications. Wireless Netw, 2013, 19: 2029–2047
23 Yang G, Wang R, Zhao K, et al. Queueing analysis of DTN protocols in deep-space communications. IEEE Aerosp Electron Syst Mag, 2018, 33: 40–48
24 Yang G, Wang R, Burleigh S C, et al. Analysis of Licklider transmission protocol for reliable file delivery in space vehicle communications with random link interruptions. IEEE Trans Veh Technol, 2019, 68: 3919–3932
25 Bisacchi A, Caini C, de Cola T. Multicolor Licklider transmission protocol: an LTP version for future interplanetary links. IEEE Trans Aerosp Electron Syst, 2022, 58: 3859–3869
26 Koo C H, Burleigh S C. Aggressive and proactive LTP control signal handling for minimal session delivery time: RTT rules the world. J Commun Netw, 2023, 25: 516–531
27 Zhao K, Wang R, Burleigh S C, et al. Modeling memory-variation dynamics for the Licklider transmission protocol in deep-space communications. IEEE Trans Aerosp Electron Syst, 2015, 51: 2510–2524
28 Yang Z, Wang R, Yu Q, et al. Analytical characterization of Licklider transmission protocol (LTP) in cislunar communications. IEEE Trans Aerosp Electron Syst, 2014, 50: 2019–2031
29 Bezirgiannidis N, Tsaoussidis V. Packet size and DTN transport service: evaluation on a DTN Testbed. In: Proceedings of International Congress on Ultra Modern Telecommunications and Control Systems, Moscow, 2010. 1198–1205
30 Lu H, Jiang F, Wu J, et al. Performance improvement in DTNs by packet size optimization. IEEE Trans Aerosp Electron Syst, 2015, 51: 2987–3000
31 Lent R. Analysis of the block delivery time of the Licklider transmission protocol. IEEE Trans Commun, 2019, 67: 518–526
32 Lent R. Learning the optimal LTP segment size for minimal turnaround times. In: Proceedings of IEEE International Conference on Communications, Seoul, 2022, 4703–4708
33 Zhou Y, Wang R, Zhao K, et al. A study of cross-layer BP/LTP data block size in space vehicle communications over lossy and highly asymmetric channels. IEEE Trans Veh Technol, 2020, 69: 16126–16141
34 Fan F, Meng H, Hu B, et al. Roulette wheel balancing algorithm with dynamic flowlet switching for multipath datacenter networks. IEEE ACM Trans Networking, 2021, 29: 834–847
35 Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance. Phys A-Stat Mech its Appl, 2012, 391: 2193–2196
36 Dowdy S, Wearden S, Chilko D. Statistics for Research. 3rd ed. 2004. Hoboken: John Wiley & Sons, Inc.
37 Yu Q, Wang R, Zhao K, et al. Modeling RTT for DTN protocol over asymmetric cislunar space channels. IEEE Syst J, 2016, 10: 556–567
38 Burleigh S C. Interplanetary overlay network design and operation v4.1.1. JPL D-48259, NASA's Jet Propulsion Laboratory (JPL), 2022