Performance Modeling of Licklider Transmission Protocol (LTP) in Deep-Space Communication

QIAN YU
Soochow University
Suzhou, China

SCOTT C. BURLEIGH
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA

RUHAI WANG
Lamar University
Beaumont, TX, USA
and
Soochow University
Suzhou, China

KANGLIAN ZHAO
Nanjing University
Nanjing, China

Delay/disruption tolerant networking (DTN) offers a solution to communications in "challenged" networks. Some work has been seen in evaluating the performance of DTN protocols based on simulated or emulated file transfer experiments. However, there is a need for a model of the performance of the DTN Licklider transmission protocol (LTP), which particularly targets reliable data transmission in deep space. In this paper, we present a performance model of LTP-based DTN data transmission in challenging communications characterized by extremely long signal propagation delay, lengthy link disruptions, and highly lossy channels that are typical of deep-space links. The model is verified by file-transfer experiments using a PC-based testbed.

Manuscript received November 26, 2013; revised May 21, 2014, September 20, 2014; released for publication September 28, 2014.

DOI. No. 10.1109/TAES.2014.130763.

Refereeing of this contribution was handled by T. Robertazzi.

This work was supported in part by the Future Networks Innovation Institute of Jiangsu Province, China, for a Prospective Research Project on Future Networks under Grant BY2013039-3-10 and by the National Natural Science Foundation (NSFC) of China under Grant 61401194.

Authors' addresses: Q. Yu, School of Electronics and Information Engineering, Soochow University, No. 1 Shizi Street, Suzhou, Jiangsu Province 215006, P. R. China; S. C. Burleigh, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109; R. Wang, Phillip M. Drayer Department of Electrical Engineering, Lamar University, 211 Redbird Lane, Beaumont, TX 77710-1029. K. Zhao, School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu Province 210093, P. R. China. Corresponding author is Kanglian Zhao, E-mail: (zhaokanglian@nju.edu.cn)

I. INTRODUCTION

The National Aeronautics and Space Administration (NASA) is developing a space internet-working architecture (or simply space Internet) that automates mission communications to enable as much autonomous operation as possible. The communication conditions in deep space violate critical assumptions underlying protocols designed for the terrestrial Internet, i.e., TCP/IP [1–5]. A large number of alternative networking architectures and protocols have been developed for implementing the space Internet [1, 6–9]. Some surveys and tutorials are also available on the development of space Internet [10–13].

Delay/disruption tolerant networking (DTN) architecture [9, 14] has been proposed as the basis for the space Internet. It is currently considered by NASA the only candidate protocol suite that approaches the level of maturity required to handle the disconnection and delays inherent in space communications [15].

The DTN architecture is designed to operate as an overlay above the transport layers of the networks it interconnects. Its operation relies heavily on a new protocol, bundle protocol (BP) [16], to form a store-and-forward overlay network for DTN. BP does no transmission directly but rather utilizes the transmission protocols of the underlying networks, which are termed "convergence layer" protocols. In order to transmit data by means of a convergence layer protocol, BP invokes the services of an interface called a "convergence layer adapter" (CLA) [16], which in turn operates the convergence layer data transport protocol stack. Currently, the TCP-based CLA (i.e., TCPCL adapter) [17], user datagram protocol (UDP)-based CLA (i.e., UDPCL adapter) [18], Saratoga CLA [19], and Licklider transmission protocol (LTP) [20] CLA (or simply LTPCL adapter [21]) are broadly supported under BP. Among these CLAs, the LTPCL adapter [20] is, in particular, designed to operate over long-haul, deep-space radio frequency links or similar links characterized by an extremely long propagation delay and/or frequent interruptions in connectivity.

The performance of DTN (mainly TCPCL and LTPCL) convergence-layer protocols in space communication systems has previously been studied by simulation or emulation file transfer experiments. However, most of these studies focus on Earth-orbit and cislunar communication scenarios. Because these scenarios are characterized by signal propagation delays that are orders of magnitude shorter than the propagation delays in deep-space communications, conclusions drawn from these studies are most likely not applicable to deep-space scenarios. More importantly, while these studies yielded valuable experimental results, those results were not supported by any theoretical analysis.

In the interest of being able to anticipate the performance of DTN protocols more accurately, we became interested in building a deterministic analytical model. While various analytical models have been built to study TCPCL-based traffic, there is currently no model that can be used to evaluate the performance of LTPCL in an analytical manner. A theoretical model would enable better understanding of the operation and the performance of LTPCL in deep-space communications. That is the motivation of this work.

In this paper, we present a performance model of LTP-based data transmission in a typical relay-based deep-space communication system characterized by extremely long signal propagation delay, lengthy link disruptions, and lossy data links. The LTP-based transmission is conducted by running the LTP protocol stack, BP/LTPCL/UDP/IP, which is simply called LTPCL for the rest of this paper.

In Section II, related work is described. The contributions of the current work are also presented in Section II. An overview of a typical DTN protocol stack for relay-based deep-space communication is presented in Section III. A model is built for the file delivery time of LTP-based data transmission in Section IV. Experimental performance evaluation results are presented and compared with predictions from the model in Section V. Conclusions are drawn in Section VI.

II. RELATED WORK AND CONTRIBUTIONS

A. Related Work and Differences of Our Work

A lot of work has been done in studying DTN protocols and techniques for wireless mobile and sensor-based networks, with most studies focusing on routing issues in DTN. In [14], Zhang presented a comprehensive overview of this work focusing on routing and discussed the challenges of the DTN technology. Some experimental work has recently been done to verify the effectiveness of the DTN architecture and protocols in satellite and space (including deep-space) communications. In [22], Caini et al. introduced DTN also as an alternative solution to Earth-orbit satellite communications. In [23], Ramadas proposed a two-dimensional priority paradigm aimed at optimizing overall data communication performance and integrated this priority paradigm with LTP. A series of Deep Impact Network Experiments (DINETs) [24] was recently conducted by the Jet Propulsion Laboratory (JPL), using the EPOXI spacecraft as the first DTN router in space. The DINET project proved the feasibility of exercising DTN protocols in deep-space environment but did not include a formal performance evaluation in a theoretical manner.

A series of studies [25–27] have been done in space networking protocol development and evaluation by Wang et al. for application to Earth-orbit and cislunar communications. Wang's group recently also did some work on experimental performance evaluation of DTN protocols, mainly on TCPCL and LTPCL, in space communications [28–31]. However, all of these studies target cislunar communications. As mentioned, the research findings from these studies may not be applicable to deep-space communications because the propagation delays of Earth-orbit and cislunar communications are orders of magnitude shorter than the propagation delays in deep-space communications. In addition, these studies focus on experimental evaluation using a testbed without solid support based on theoretical analysis.

Some analytical work has also been done in estimating the file delivery time of recently developed transmission protocols targeting deep-space communications. In [32], Bezirgiannidis et al. presented a method of estimating data bundle delivery time in space internet working using the contact graph routing (CGR) route computation algorithm. Unlike our work focusing on LTP, the analysis in [32] is done at the BP layer focusing on estimation of end-to-end bundle delivery time rather than hop-to-hop LTP data block transmission time. Also, the study in [32] adopts the natural approach of "one bundle per block" for analysis and does not take into consideration the effect of bundle aggregation. However, our prior study [30, 31] shows that bundle aggregation significantly improves BP/LTPCL performance over highly asymmetric channels, so we exercise that feature in our work with highly asymmetric space channel data rates. In addition, both the analysis and experimental verification in [32] are done for space communication scenarios with quite short signal propagation delay (up to 100 s); these research findings are not directly applicable to the very deep space communications which we target. Moreover, neither long link disruptions nor severe channel-rate asymmetry is considered in the analytical performance formulation and experiments of bundle transmission in [32]. Finally, only quite low channel bit error rates (BERs) (up to 2.42×10^{-7}) are examined in [32]; these again do not characterize very deep space communications.

A theoretical analysis of the expected file delivery time of Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) in deferred negative acknowledgment (NAK) mode over a single hop is presented in [33]. Although the deferred NAK operation mode of CFDP is functionally similar to the retransmission procedures defined for LTP, CFDP is somewhat different from the BP/LTPCL stack. Unlike LTP, for example, CFDP in deferred mode requires transmission of a meta-data protocol data unit (PDU) prior to data PDU transmission and transmission of a separate end of file (EOF) PDU following the end of data PDU transmission. These differences of operation subtly affect the file delivery performance of the two protocols, causing the formula derived for the expected total number of transmission efforts in this work to be slightly different from the one derived in [33] [i.e., (8)]. Also, as in the case of the work presented in [32], link disruption and channel-rate asymmetry are considered in neither the analytical formulation nor the performance simulation of CFDP in [33]. In addition, the analysis in [33] is not validated by file delivery experiments based on an experimental infrastructure or a testbed, as in our work.

B. Contributions

This work focuses on performance modeling of DTN architecture and LTP protocols in challenging deep-space communications, analytically studying the performance (mainly file delivery time) of DTN's LTPCL for reliable data transmission over deep-space channels. File transfer experiments are conducted using a PC-based testbed to validate the model and strengthen the analysis of the protocol. The model is useful in understanding the performance of LTP-based protocols in a file transfer scenario over a deep-space channel, an environment that is difficult to study in practice. These are the main contributions of this paper.

We note that the derivation of most of the equations that model the behavior mandated by the LTP specification is straightforward. Their validity is confirmed by the fact that [32] derived essentially similar formulations, working independently. Similar equations were also developed in [33] which considered a different protocol, CFDP, whose "acknowledged mode procedures" are functionally similar to – indeed, were the model for – the retransmission procedures defined for LTP.

III. TYPICAL DEEP-SPACE IPN COMMUNICATIONS ARCHITECTURE AND DTN PROTOCOLS FOR SPACE

To improve data transmission efficiency, relay-based transmission has been adopted in deep-space communications. This has enabled recent Mars exploration missions to transmit a high volume of science data. In [34, Fig. 2], a nominal relay-based deep-space Interplanetary Internet (IPN) communication architecture and corresponding DTN protocol reference stack are illustrated.

As described in [16], a store-and-forward transmission mechanism and a custody transfer option are the main techniques of BP for dealing with the link disconnections and long delays inherent in space and other challenging communications. For a detailed discussion of these two mechanisms and their operation in a deep-space scenario, see [16, 34].

As mentioned, we focus primarily on LTPCL because it was specifically designed to provide reliability in communications over deep-space links. In fact, both reliable and unreliable transmission service can be provided for a single client data block [20]: a single block can contain a data part for which reliable data transfer is required (termed the "red" part) followed by a data part for which reliable data transfer is unnecessary (the "green" part). The reliable delivery of the "red" part of the block is assured through LTP's own acknowledgment and retransmission mechanisms. For LTPCL, BP passes bundles as "service data units" to LTP for transmission; LTP encapsulates service data units in LTP blocks. Each block is fragmented into LTP data segments, according to the maximum transmission unit (MTU) size of the link service that underlies LTP; each segment is annotated with both its length and its displacement from the start of the block. The last segment of each LTP data block that contains only red data is flagged as a checkpoint (CP) to elicit a reception report and also as an end of block (EOB) to indicate total block length. When a CP is received, the receiver returns a report of cumulative reception for that block, termed a report segment (RS), i.e., an ACK for the block. In other words, an RS acknowledges a CP, and it serves either as a positive acknowledgment (if all data were received) or a negative acknowledgment (if some data were not successfully received and must be retransmitted). The sender returns a report acknowledgment (RA) to the receiver in response to an RS. Both RSs and CPs are on timers, and they are retransmitted if not acknowledged.

The communication channels in space are frequently asymmetric in terms of channel rate [35]. To reduce the effect of space channel-rate asymmetry on transmission effectiveness, LTP is designed so that it may aggregate multiple small BP bundles into a single block; this can reduce the number of ACKs on the uplink. As mentioned, our prior experimental study shows that bundle aggregation significantly improves LTP performance over highly asymmetric channels (although it is considered nonbeneficial in other work [36]). Therefore, LTP's bundles aggregation mechanism is exercised in the current work for both analytical modeling and experimental verification.

IV. PERFORMANCE MODELING OF LTP-BASED TRANSMISSION

In this section, we build an analytical model for the performance of LTPCL in deep-space communication scenarios. The model is built to predict file delivery latency based on the expected total number of transmission efforts required for successful delivery of an entire file. With the file delivery latency predicted, the goodput performance for file transmission, if necessary, can be easily computed by taking its ratio with the file size. The file delivery latency is defined as the time needed to complete the delivery of an entire file to its destination; this generally includes the transmission time, propagation time, processing time, and queue time.

LTP data transmission is organized in "sessions." A session is defined as the sequence of LTP segment exchanges – transmission efforts – undertaken to effect the successful transmission of a single data block. Each transmission effort is enacted by sending a sequence of data segments (initially, all segments in the block, regardless of the number), the last of which is flagged as a CP, and data transmission is acknowledged on a CP basis. This means that retransmission of all the lost segments of a block is also organized as an additional CP-delimited transmission effort which results in a new RS. This process is repeated until the whole block of data is successfully delivered. For a relatively small number of LTP blocks transmitted over a relatively high-rate

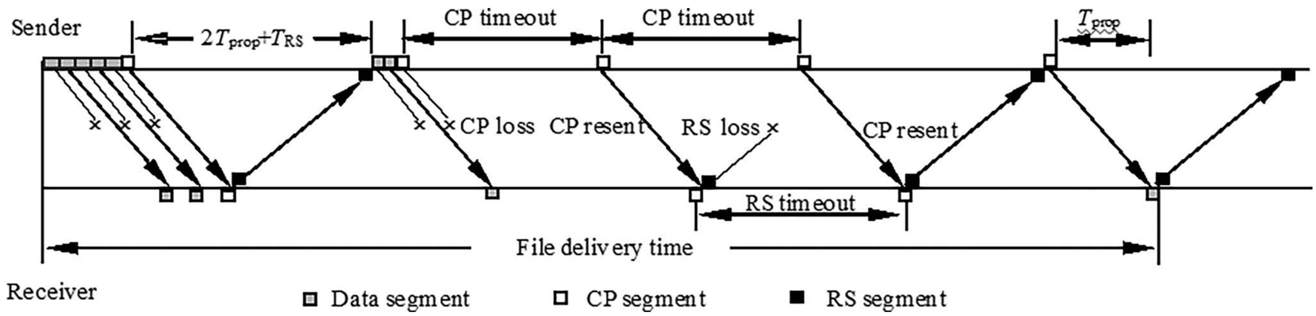

Fig. 1. LTP data blocks/segments transmission and interactions between sender and receiver.

deep-space channel with one-way propagation delay of 10 min, the propagation delay is so much greater than the time required to radiate all the data of each block that the blocks are in effect transmitted in parallel.

Fig. 1 illustrates a scenario of LTPCL data and RS segment exchange for transmission of a file in a single data block between sender and receiver. In this scenario, a block of six data segments (five regular segments and one CP segment) are transmitted in the first transmission effort. Three of the six data segments are lost due to channel noise and cannot be delivered to the receiver. After the CP is received, the receiver returns an RS in response to it, requesting retransmission of three lost data segments. The three lost segments are retransmitted (with two as regular segments and the last as another CP) in the second transmission effort, but this time the first retransmitted segment and the CP are lost. Because the CP is not received by the receiver, no RS is returned in response to it. As a result, the sender has to resend the CP after the CP timer expires (i.e., CP timeout). The re-sent CP arrives successfully at the receiver. The RS requesting retransmission of the other segment lost during the second transmission effort is returned but lost, causing another CP timeout and causing the CP to be re-sent again. In other words, the second CP timeout and retransmission is actually caused by the loss of an RS while the first one is caused by the loss of the CP itself. Moreover, the report acknowledgment that is normally sent upon receipt of an RS is of course not sent, so the RS timer expires as well, causing the RS to be retransmitted. When the re-sent RS is received by the sender, the other data segment lost during the second transmission effort is re-sent. It is re-sent as a CP segment because it is the only data segment to be sent at this time. When it arrives at the receiver, the entire block of six segments is successfully delivered to the receiver, indicating the end of the time required to accomplish the file delivery. The transmission of the last RS in response to the arrival of the last data (CP) segment is generally not considered as part of the file delivery time, because the file has already been made available to the application.

We aim to compute the expected total number of data segment transmission efforts $E[NE_{\text{seg}}]$ needed for successful delivery of an entire file so that the total file delivery time can be estimated. Let the uncorrected BER of a deep-space channel be p and the length of a data segment (or simply, a packet) be L_{seg}. Provided that all bits are transmitted independently, the loss probability of a segment can be calculated as follows:

$$p_{\text{seg}} = 1 - (1-p)^{8\times L_{\text{seg}}}$$

For data transmission over a cis-Martian communication channel, the signal propagation delay T_{prop} ranges between 4 and 20 min. If we assume commonly available data link rates and typical source file sizes in deep-space missions, all data bytes of the file are already sent out and "in flight" on the channel within a round-trip time (RTT) of 10 min. Therefore, the transmission of the file can generally be considered a single continuous sequence of data bytes. Given the entire file size of L_{File}, an aggregated LTP block size of L_{block} and the length of a data segment of L_{seg}, the total number of blocks that need to be transmitted can be represented as $L_{\text{File}}/\text{L}_{\text{block}}$ and the number of segments contained in one block can be represented as $L_{\text{block}}/L_{\text{seg}}$. The number of data segments lost during the first transmission effort $N_{\text{seg\_loss\_1}}$ can be formulated as

$$N_{\text{seg\_loss\_1}} = \frac{L_{\text{File}}}{L_{\text{block}}} \times \left(\frac{L_{\text{block}}}{L_{\text{seg}}} - 1\right) \times p_{\text{seg}}$$

where $(L_{\text{block}}/L_{\text{seg}}-1)$ represents the number of data segments that are generated from a single LTP block (the last of which may be smaller than L_{seg}) but with the last segment (i.e., the CP segment) excluded.

The segments lost in the first transmission effort need to be retransmitted in the second effort, and the number of retransmitted segments is $N_{\text{seg\_loss\_1}}$. But some of the retransmitted segments may be lost again in the second transmission effort. The number of data segments lost during the second transmission effort, defined as $N_{\text{seg\_loss\_2}}$, can be formulated as

$$N_{\text{seg\_loss\_2}} = \frac{L_{\text{File}}}{L_{\text{block}}} \times \left(\frac{L_{\text{block}}}{L_{\text{seg}}} - 1\right) \times p_{\text{seg}}^2$$

If the transmission of a source file experiences k transmission efforts (including the initial transmission and all the resulted retransmission efforts for the lost data), the number of data segments that are still lost after the k_{th} transmission effort is

$$N_{\text{seg\_loss\_}k} = \frac{L_{\text{File}}}{L_{\text{block}}} \times \left(\frac{L_{\text{block}}}{L_{\text{seg}}} - 1\right) \times p_{\text{seg}}^k \tag{1}$$

Data delivery of the source file is considered successful if the number of lost segments after the entire transmission process of k efforts (i.e., during the k_{th} transmission effort) is fewer than 1. That is, following (1),

$$\frac{L_{\text{File}}}{L_{\text{block}}} \times \left(\frac{L_{\text{block}}}{L_{\text{seg}}} - 1\right) \times p_{\text{seg}}^{k} < 1$$

In other words,

$$p_{\text{seg}}^{k} < \frac{L_{\text{block}} \times L_{\text{seg}}}{L_{\text{File}} \times \left(L_{\text{block}} - L_{\text{seg}}\right)}$$

Then, k can be derived as

$$k > \log_{p_{\text{seg}}}^{\frac{L_{\text{block}} \times L_{\text{seg}}}{L_{\text{File}} \times (L_{\text{block}} - L_{\text{seg}})}}$$

Therefore, the expected number of transmission efforts k that can ensure successful delivery of the entire file is given by

$$k = \left\lceil \log_{p_{\text{seg}}}^{\frac{L_{\text{block}} \times L_{\text{seg}}}{L_{\text{File}} \times (L_{\text{block}} - L_{\text{seg}})}} \right\rceil$$

The derived expected number of transmission efforts k is actually the expected total number of data transmission efforts needed for successful delivery of the entire file, written as $E[NE_{\text{seg}}]$, i.e.,

$$E[NE_{\text{seg}}] = k = \left\lceil \log_{p_{\text{seg}}}^{\frac{L_{\text{block}} \times L_{\text{seg}}}{L_{\text{File}} \times (L_{\text{block}} - L_{\text{seg}})}} \right\rceil \tag{2}$$

With $E[NE_{\text{seg}}]$ derived, the total file delivery time can be estimated. Denote the transmission time of all data blocks or segments (including all possible retransmissions in different transmission efforts) as T_{trans}; denote the total time that elapses during propagation of all the data blocks and corresponding RSs as $T_{\text{prop\_total}}$; denote the total time that elapses while awaiting expiration of the CP timer as $T_{\text{CP\_total}}$; denote the total time that elapses while awaiting expiration of the RS timer as $T_{\text{RS\_total}}$; and denote the total additional time lapse caused by possible link disruption (or break) as T_{break}. If the processing delay and queue delay are ignored, the expected file delivery time T_{File} of LTPCL can be formulated as

$$T_{\text{File}} = T_{\text{trans}} + T_{\text{prop\_total}} + T_{\text{CP\_total}} + T_{\text{RS\_total}} + T_{\text{break}} \tag{3}$$

Note that the second CP timeout interval observed in Fig. 1 is actually caused by a loss of its corresponding RS which also results in an RS timeout at the receiver. Only one of these timeout intervals is actually counted in (3). As mentioned, the time involved for transmission and propagation of an ACK segment (i.e., an RS for LTP) acknowledging the arrival of the last set of data segment(s) is not included in the file delivery time. Each of the time contributors to T_{File} in (3) (except T_{break}) is evaluated below. Notations we use for the evaluation are specified in Table I.

1) Transmission time of all data blocks (including all possible retransmissions of those blocks), T_{trans}

The expected number of segment transmissions needed for successful delivery of a single segment is $1/(1-p_{\text{seg}})$. Given the entire file size of L_{File} and the length of a data segment of L_{seg}, the total number of segments that need to be transmitted to ensure successful delivery of the entire file N can be formulated as

$$N = \frac{L_{\text{File}}}{L_{\text{seg}}} \left(\frac{1}{1 - p_{\text{seg}}}\right)$$

Provided that the average segment transmission time is $L_{\text{seg}}/R_{\text{Data}}$, the transmission time of all data blocks T_{trans} can be derived as

$$\begin{aligned} T_{\text{trans}} &= \frac{L_{\text{seg}}}{R_{\text{Data}}} \times N = \frac{L_{\text{seg}}}{R_{\text{Data}}} \times \frac{L_{\text{File}}}{L_{\text{seg}}} \left(\frac{1}{1 - p_{\text{seg}}}\right) \\ &= \frac{L_{\text{File}}}{R_{\text{Data}}} \left(\frac{1}{1 - p_{\text{seg}}}\right) \end{aligned} \tag{4}$$

2) Total propagation time for all data blocks and corresponding RSs, $T_{\text{prop\_total}}$

Similar to the transmission of data segments, under the assumption of independent bit errors, the probability that a given block is successfully transmitted is independent from the transmission success probability of all other blocks. Therefore, the expected number of transmission efforts for a block $E[NE_{\text{seg}}]$ is actually the expected number of transmission efforts for the entire file. In this case, the total propagation time $T_{\text{prop\_total}}$ of the entire file resulting from $E[NE_{\text{seg}}]$ data transmission efforts is

$$T_{\text{prop\_total}} = \left(E\left[NE_{\text{seg}}\right] - 1\right)\left(2T_{\text{prop}} + T_{\text{RS}}\right) + T_{\text{prop}} \tag{5}$$

where $(E[NE_{\text{seg}}]-1)$ is the expected number of transmission efforts excluding the final one and the additional T_{prop} is the propagation time for the final transmission effort of data segments.

3) Total time that elapses while awaiting timeout of CP timer, $T_{\text{CP\_total}}$

Define the timeout of CP segment timer as $T_{\text{CP\_timer}}$. For the best transmission efficiency, it is generally reasonable to define it as

$$T_{\text{CP\_timer}} = 2T_{\text{prop}} + T_{\text{RS}}$$

TABLE I
Notations in Performance Modeling

Symbol	Definition
p	Probability of channel bit error
p_{seg}	Probability of error in delivering a data segment
p_{RS}	Probability of error in delivering an RS segment
p_{CP}	Probability of error in delivering a CP segment
$E[NE_{\text{seg}}]$	Expected number of data transmissions efforts for successfully delivering a file
T_{prop}	One-way propagation delay
T_{RS}	Transmission time of an RS segment
T_{CP}	Transmission time of a CP segment
$T_{\text{RS\_timer}}$	Timeout length of RS timer
$T_{\text{CP\_timer}}$	Timeout length of CP timer
T_{trans}	Transmission time of all data blocks (including all possible retransmissions of them distributed in different transmission efforts)
$T_{\text{RS\_total}}$	Total time caused by possible timeout of RS timer
$T_{\text{CP\_total}}$	Total time caused by possible timeout of CP timer
T_{File}	File delivery time

Assuming that the transmission of CP segments conforms to a geometrical distribution, the expected transmission efforts for a CP segment can be written as

$$E[NE_{\mathrm{CP}}] = \frac{1}{1 - p_{\mathrm{CP}}}$$

where p_{CP} is the loss probability of a CP segment with length of L_{CP}, and similar to p_{seg}, it is formulated as

$$p_{\mathrm{CP}} = 1 - (1 - p)^{8\times L_{\mathrm{CP}}}$$

Then, the total expected time that elapses while awaiting timeout of CP timer $T_{\mathrm{CP\_total}}$ is given as

$$\begin{aligned} &T_{\mathrm{CP\_total}} \\ &= (E[NE_{\mathrm{CP}}] - 1)(T_{\mathrm{CP}} + T_{\mathrm{CP\_timer}})\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \\ &= \frac{p_{\mathrm{CP}}}{1 - p_{\mathrm{CP}}}(T_{\mathrm{CP}} + T_{\mathrm{CP\_timer}})\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \\ &= \frac{p_{\mathrm{CP}}}{1 - p_{\mathrm{CP}}}\left(T_{\mathrm{CP}} + 2T_{\mathrm{prop}} + T_{\mathrm{RS}}\right)\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \end{aligned} \tag{6}$$

4) Total time that elapses while awaiting timeout of RS timer, $T_{\mathrm{RS\_total}}$

Similar to the discussion of $T_{\mathrm{CP\_total}}$, the total expected time that elapses while awaiting timeout of the RS timer, defined as $T_{\mathrm{RS\_total}}$, can be derived as

$$\begin{aligned} &T_{\mathrm{RS\_total}} \\ &= \left(\frac{1}{1 - p_{\mathrm{RS}}} - 1\right)(T_{\mathrm{RS}} + T_{\mathrm{RS\_timer}})\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \\ &= \frac{p_{\mathrm{RS}}}{1 - p_{\mathrm{RS}}}\left(T_{\mathrm{RS}} + 2T_{\mathrm{prop}} + T_{\mathrm{CP}}\right)\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \end{aligned} \tag{7}$$

where p_{RS} is the loss probability of an RS segment with length of L_{RS}, formulated as $p_{\mathrm{RS}} = 1 - (1 - p)^{8\times L_{\mathrm{RS}}}$.

Plugging (4), (5), (6), and (7) into (3), the expected file delivery time for LTP transmission over a deep-space communication channel can be represented as

$$\begin{aligned} &T_{\mathrm{File}} \\ &= T_{\mathrm{trans}} + T_{\mathrm{prop\_total}} + T_{\mathrm{CP\_total}} + T_{\mathrm{RS\_total}} + T_{\mathrm{break}} \\ &= \frac{L_{\mathrm{File}}}{R_{\mathrm{Data}}}\left(\frac{1}{1 - p_{\mathrm{seg}}}\right) + \left(E\left[NE_{\mathrm{seg}}\right] - 1\right)\left(2T_{\mathrm{prop}} + T_{\mathrm{RS}}\right) + T_{\mathrm{prop}} \\ &\quad + \frac{p_{\mathrm{CP}}}{1 - p_{\mathrm{CP}}}\left(T_{\mathrm{CP}} + 2T_{\mathrm{prop}} + T_{\mathrm{RS}}\right)\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) \\ &\quad + \frac{p_{\mathrm{RS}}}{1 - p_{\mathrm{RS}}}\left(T_{\mathrm{RS}} + 2T_{\mathrm{prop}} + T_{\mathrm{CP}}\right)\left(E\left[NE_{\mathrm{seg}}\right] - 1\right) + T_{\mathrm{break}} \end{aligned} \tag{8}$$

Therefore, in some deep-space communication scenarios for which goodput performance is needed as a measure of link utilization for the LTP protocol stack, it may be derived as

$$\gamma = \frac{L_{\mathrm{File}}}{T_{\mathrm{File}}} \tag{9}$$

V. EXPERIMENTAL EVALUATION AND MODEL VALIDATION

In this section, we present an experimental performance evaluation of LTPCL based on file transfer experiments over emulated deep-space channels, focusing on a comparison between the file delivery time predicted by the model and the observed delivery time, in validation of the model. Due to space constraints, only a few sets of representative comparison results are presented. The experimental setup and configurations are described prior to presentation of the experimental results.

TABLE II
Experimental Factors and Configuration

Experimental Factors	Settings/Values
DTN protocol implementations	Interplanetary Overlay Network (ION) v3.2.1 from Jet Propulsion Laboratory (JPL), Caltech
DTN protocol layering and configurations	BP/LTPCL/UDP/IP
BP custody transfer option	Enable
LTP red/green settings	Bundles are set 100% red data
MTU size	1500 bytes
LTP segment size	1400 bytes
Bundle size	40000 bytes
Link disruption (hour)	0, 0.25, 0.5, 1, 2, 4, and 8
Buffer size	70 Mbytes
Channel ratio (data rate : ACK rate)	500/1 (2 Mbits/s : 4 Kbits/s)
BER	10^{-7}, 10^{-6}, 5×10^{-6}, and 10^{-5}
One-way link delay	600 s (10 min)
Experimental file size	600 Kbytes, 6 Mbytes, 60 Mbytes

A. Experimental Setup and Configurations

The file transfer experiments were conducted by running the DTN protocols over a deep-space communication emulation infrastructure. A PC-based space communication and networking testbed (SCNT) was built to implement an emulated deep-space communications infrastructure. Please refer to [28, Fig. 1] for a block diagram of the SCNT testbed. The testbed consists of a deep-space link simulation box and three Linux-based PCs—a source PC called TX, a relay PC called MX, and a destination PC called RX. A file transfer from TX, through MX, to RX emulates data transfer from Mars to the Earth through a Mars orbiter. Previous work [27, 28, 31] shows that the evaluation results obtained from the SCNT have generality and the testbed can effectively evaluate the performance of a protocol.

The file transfer experiment with LTPCL was conducted by running BP/LTPCL over UDP/IP on the SCNT, i.e., running LTPCL. The DTN BP and LTPCL protocol implementations used for our experiments were provided by the Interplanetary Overlay Network (ION) distribution v3.2.1 [37] developed by the JPL, California Institute of Technology. For LTPCL, the BP bundle was declared to be 100% red data to ensure reliable transmission of all data bytes.

Table II lists the major configuration parameters of our experiments. The experiments were conducted by transferring text files of three sizes – 600 Kbytes, 6 Mbytes, and 60 Mbytes – using the SCNT testbed.

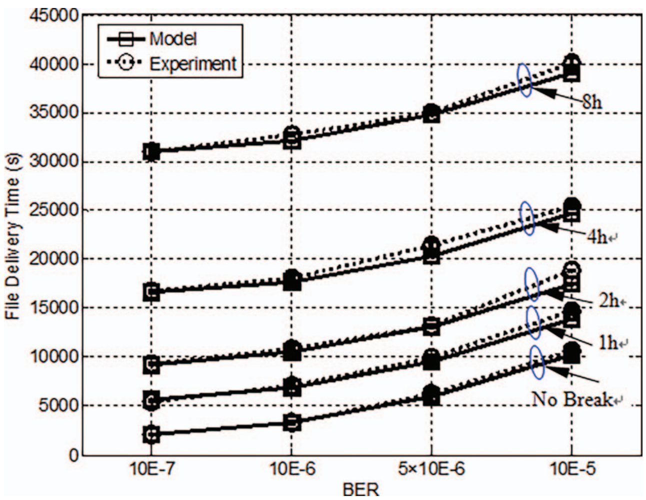

Fig. 2. Comparison of file delivery time in transmission of 60 Mbytes file predicted by the model and observed in experimental BP/LTP transmissions over a deep-space channel characterized by one-way link delay of 10 min at four BERs, with link disruption durations of 0 h (no disruption), 1 h, 2 h, 4 h, and 8 h.

Sixteen test runs were performed in each SCNT configuration. The source file was divided into fragments of 40k bytes each for transmission as the payloads of bundles. BP passes its bundles as "service data units" to LTPCL for transmission. The service data units are aggregated into LTP client data "blocks"; the block size was configured to be 244 Kbytes, large enough to contain 6 complete service data units. Considering that space communications channels are generally characterized by asymmetric channel rates, the effect of asymmetric channel rates is also taken into consideration in the file transfer experimental evaluation. A channel ratio is defined as a ratio of data channel rate over the ACK channel rate. It is well known that the Internet protocol generally cannot work effectively when the channel ratio is higher than 50/1 [38]. Based on this information, a channel ratio of 500/1 was configured in the experiment, with 2 Mbits/s for the data channel and 4 Kbits/s for the ACK channel.

B. Performance Comparison between Model and Experiments

In Fig. 2, the file delivery time in transmission of a 60 Mbyte file predicted by the model [i.e., (8)] is compared with that observed in experimental LTPCL transmissions over a deep-space channel, with link disruption durations of 0 h (no disruption), 1 h, 2 h, 4 h, and 8 h. For each of five link disruption scenarios, as the channel BER increases, the file delivery time of both the model and the experiments increases. However, the differences of the file delivery time between any two given link disruption scenarios are roughly equal for all the BERs. This indicates that the effect of lengthy link disruption dominates the transmissions, and variation of deep-space channel noise level does not significantly affect the differences in file delivery time between any two link disruption scenarios.

The file delivery time predicted by the model is very close to that observed in the experiments at all channel noise levels for a given link disruption scenario. The latencies observed in the experiments are consistently either equal to or slightly higher than those predicted by the model. They are roughly equal at a low BER such as 10^{-7} or 10^{-6} and slightly higher at a high BER. This is reasonable. As mentioned in Section III, the processing time for all LTP segments (including data, RS, and RA segments) and the segment queuing time at each node were ignored for simplicity when the model was built. This additional segments-related time might cause additional delivery latency in the file transfer experiments, especially for transmissions over a lossy channel which results in more retransmissions of segments and therefore a minor increase in file delivery time.

For a better view of file delivery time of LTPCL transmissions with respect to variations of link disruption duration, Fig. 3 illustrates the file delivery time in transmission of a 60 Mbyte file with increasing link outage duration for both the model and the experiments at each of four BERs. Note that, in each link disruption scenario, the increase of file delivery time and its variation trend, together with the increase of disruption duration, are the same even for different BERs. At each of the configured BERs, the file delivery time values from both the model and the actual experiments in Fig. 3 increase roughly exponentially along with an exponential increase in link disruption from 0.25 h to 8 h. This indicates again that the lengthy link disruption in file transmission plays a dominant role in the file delivery latency.

With respect to a comparison between the model and the experiments, the file delivery time values measured from the actual experiment match those of the model very well. This is true for all channel noise levels. This is an indication that the model is valid whether or not the link is disrupted and that it can accurately quantize the effects of link delay, link disruption, and channel-rate asymmetry regardless of channel (noise) quality.

To study the effect of variation in file size, Fig. 4 presents a comparison of file delivery time for files of 600 Kbytes, 6 Mbytes, and 60 Mbytes at each of four BERs without link disruption for both the model and the experiments. A direct observation is that for all three file sizes, the file delivery time increases along with the increase of channel BER. This is reasonable because, statistically, no matter the file size, a higher channel BER results in a larger number of data corruption events during data transmission. This calls for more retransmission events and very likely more transmission cycles, thereby resulting in an increase in the file delivery time.

Another important observation from Fig. 4 is that for the transmissions over the channel with a relatively low loss rate (i.e., with a BER of 5×10^{-6} or lower), the file delivery times of all three files are consistently very close—the files of 600 Kbytes and 6 Mbytes have almost

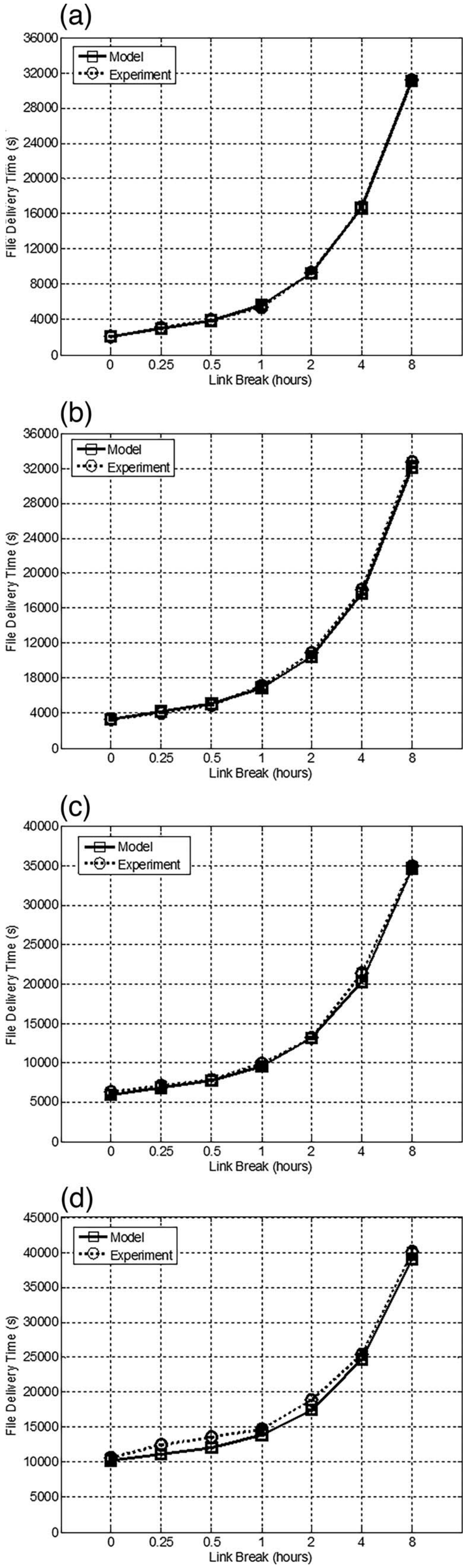

Fig. 3. Comparison of file delivery time between model and experiments for BP/LTP transmission of 60 Mbytes file over deep-space channel characterized by one-way link delay of 10 min with link outage of 0-8 h at each of four BERs. (a) BER $= 10^{-7}$. (b) BER $= 10^{-6}$. (c) BER $= 5 \times 10^{-6}$. (d) BER $= 10^{-5}$.

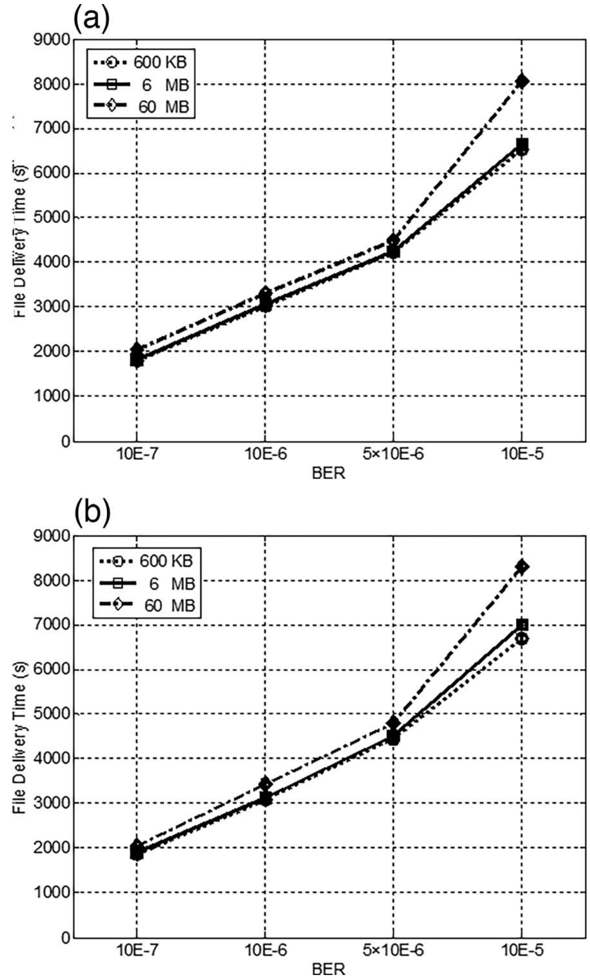

Fig. 4. Comparison of file delivery time among files of 600 Kbytes, 6 Mbytes, and 60 Mbytes for BP/LTP transmission at each of four BERs for both model and experiments. (a) Model. (b) Experiment.

the same delivery latency while the latency for the file of 60 Mbytes is slightly longer. The observation of approximately equal file delivery time among different file sizes at a relatively low loss rate is more obvious when delivery latencies are plotted against file sizes as illustrated in Fig. 5. As can be observed in Fig. 5, the file delivery time remains almost unchanged at each of the BERs of 10^{-7}, 10^{-6}, and 5×10^{-6}. However, an obvious increase of file delivery time is observed at the BER of 10^{-5} for the file size increase from 6 Mbytes to 60 Mbytes.

This result is in accord with our model. While (2) predicts an increase in the expected number of transmission efforts (and thus expected file delivery time) both with increasing BER and with increasing file size at a given BER, the increase is far more sensitive to variation in BER than to variation in file size. A file of 6 Mbytes is only 5.4 Mbytes larger than a file of 600 Kbytes. A file of 60 Mbytes, however, is 54 Mbytes larger than a file of 6 Mbytes, a difference in file size that is substantial enough to be reflected in the file delivery times noted in Fig. 5. As shown in Table III, the file delivery required two transmission cycles, three cycles, and four cycles at the

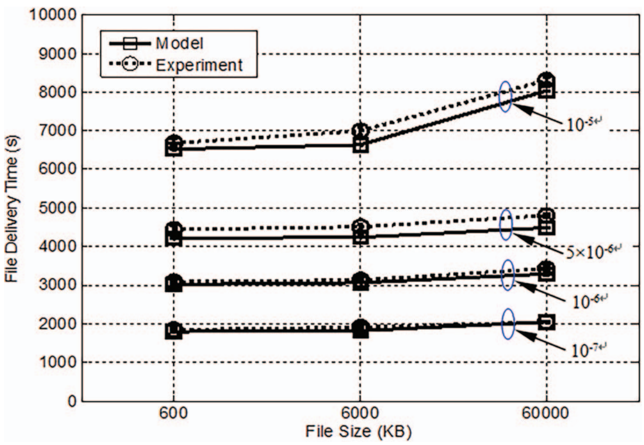

Fig. 5. Comparison of file delivery time between model and experiments for BP/LTP transmission among files of 600 Kbytes, 6 Mbytes, and 60 Mbytes over deep-space channel characterized by one-way link delay of 10 min at each of four BERs.

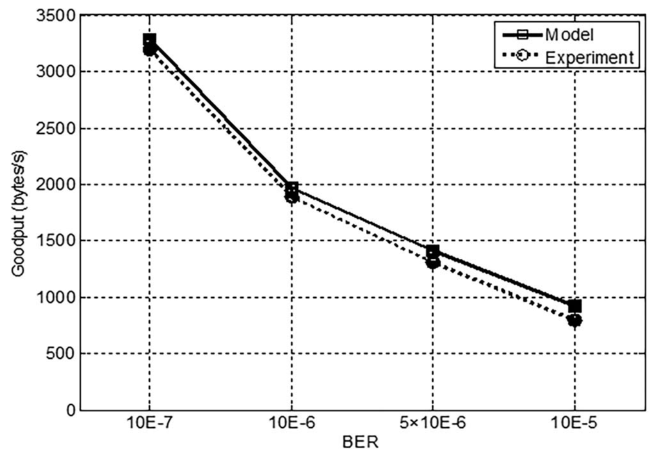

Fig. 6. Comparison of goodput between model and experiments for BP/LTP transmission of 6 Mbytes file over deep-space channel characterized by one-way link delay of 10 min at four BERs.

TABLE III
Expected Total Number of Transmission Efforts required for Successful Delivery of a File

Expected File Transmission Efforts		File Size		
		600 Kbytes	6 Mbytes	60 Mbytes
BER	10^{-7}	2	2	2
	10^{-6}	3	3	3
	5×10^{-6}	4	4	4
	10^{-5}	6	6	7

BERs of 10^{-7}, 10^{-6}, and 5×10^{-6}, respectively, regardless of file size. However, at the BER of 10^{-5}, the delivery of a file of 60 Mbytes required one more retransmission cycle than the delivery of files of the other two sizes. As a result, the file of 60 Mbytes had significant longer file delivery time than the files of 600 Kbytes and 6 Mbytes at the BER of 10^{-5}.

Note that the largest file, 60 Mbytes, is also characterized by a much longer file transmission time in comparison with the other two file sizes. Given the data channel rate of 2 Mbits/s configured in our experiments, the raw file transmission time is 2.4 s, 24 s, and 240 s for 600 Kbytes, 6 Mbytes, and 60 Mbytes, respectively. In other words, the file transmission time for a file of 60 Mbytes constitutes a much larger fraction of round-trip propagation latency, with a correspondingly more noticeable impact on the total file delivery time. This is true both for the results predicted by the model and those measured from the experiments in Fig. 4.

Note that the file of 6 Mbytes, although larger than 600 Kbytes, does not require more transmission efforts than the file of 600 Kbytes according to Table III. Therefore, the two files have approximately equal file delivery times with only negligible difference caused by the slightly longer file transmission time for the file of 6 Mbytes.

As mentioned earlier, goodput performance may be a suitable measure of link utilization in some deep-space communication scenarios, especially for continuous file transmission over continuous links. A comparison of the goodput performance for LTPCL transmission of a 6 Mbyte file over a nondisrupted link, as predicted by the model [i.e., (9)] and as observed in the experiments, is presented in Fig. 6. As channel BER increases, goodput performance drops because of the increase in file delivery time caused by additional data losses and retransmissions that result in more transmission cycles.

The model accurately predicts goodput performance and its variation trend at all four channel noise levels. Predicted and observed goodput match perfectly at low BERs from 10^{-7} to 10^{-6}. At a higher BER of 10^{-6}–10^{-5}, the goodput performance observed in the experiments is slightly lower than predicted. As discussed earlier, the time required for processing all the LTP segments and time during which segments remain in queues awaiting processing were omitted from the model. These considerations indeed have negligible impact on file delivery time for transmission over clean channels. But for transmissions over a lossy channel which results in transmission and processing of many more segments, these effects may cause an observable difference in file delivery time and thus some goodput degradation, as observed at the BERs of 10^{-6}, 5×10^{-6}, and 10^{-5} in Fig. 6. In fact, similar divergence of goodput performance is also observed between the model and the experiments for two other file sizes (600 Kbytes and 60 Mbytes) at a BER of 10^{-5}. The transmission of a 60 Mbyte file has slightly more divergence than the other two, and it even shows observable divergence at the BERs of 10^{-6} and 5×10^{-6}. This is because a large file is transmitted in more segments, which results in relatively more transmission time, processing time, and queue time that the model does not account for, and thus more divergence from the predictions of the model.

VI. CONCLUSIONS

In this paper, an analytical model has been developed for the expected file delivery time for LTP-based transmissions in a deep-space communications scenario. Experimental results based on file transfers over a PC-based testbed validate the analytical model, indicating that the model can predict the performance in data-flow file transfer in deep-space communications. According to the study based on both analytical modeling and experimental evaluations, DTN's LTPCL is effective in successfully handling file delivery over deep-space channels in the presence of an extremely long link delay, lengthy link disruptions, various data loss rates, and highly asymmetric channel rates. The experiments incorporating link disruptions of 0-8 h indicate that file delivery time increases as the channel noise level increases. However, the differences in file delivery time between any two given link disruption scenarios are roughly equal regardless of BER, indicating that the effect of lengthy link disruption dominates the transmissions, and variation of deep-space channel noise level does not significantly affect the differences in file delivery time between any two link disruption scenarios. The study also indicates that transmissions of files with significantly different sizes have approximately equal file delivery time at a low or even moderately high loss rate (with a BER $\leq 5 \times 10^{-6}$) because they result in almost equal numbers of total transmission efforts. However, for transmissions with a very high data loss rate (i.e., BER of 10^{-5}), variation in file size has obvious impact on the file delivery time. This is because of additional transmission efforts required for a large file to ensure its successful delivery, leading to significantly longer file delivery time.

VII. FUTURE WORK

This study focuses on the performance analysis of DTN's main CLA for file delivery in deep space, LTPCL working with BP. The operation of unreliable UDPCL associated with the BP custody transfer option enabled is an alternative for reliable data delivery. Performance modeling and experimental validation of UDPCL under BP with the custody transfer option enabled, in comparison with the performance of reliable LTPCL presented in this work, will be helpful in evaluating the performance of the custody transfer mechanism, and it is therefore left as future work.

ACKNOWLEDGMENT

The research described in this paper was performed in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors would like to acknowledge Dr. Thomas Robertazzi at Stony Brook University for the help in improving the technical quality of the paper.

REFERENCES

[1] Akyildiz, I. F., Morabito, G., and Palazzo, S.
TCP Peach: A new congestion control scheme for satellite IP networks.
IEEE/ACM Transactions on Networking, **9**, 3 (June 2001), 307–321.

[2] Bisio, I., and Marchese, M.
Analytical expression and performance evaluation of TCP packet loss probability over geostationary satellite.
IEEE Communications Letters, **8**, 4 (Apr. 2004), 232–234.

[3] Bisio, I., and Marchese, M.
Power saving bandwidth allocation over GEO satellite networks.
IEEE Communications Letters, **16**, 5 (May 2012), 596–599.

[4] De Cola, T., Ernst, H., and Marchese, M.
Performance analysis of CCSDS File Delivery Protocol and erasure coding techniques in deep space environments.
Computer Networks, **51**, I14 (Oct. 2007), 4032–4049.

[5] De Cola, T., and Marchese, M.
Performance analysis of data transfer protocols over space communications.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 4 (Oct. 2005), 1200–1223.

[6] Akan, O., Fang, J., and Akyildiz, I. F.
TP-Planet: A reliable transport protocol for Interplanetary Internet.
IEEE Journal on Selected Areas in Communications, **22**, 2 (Feb. 2004), 348–361.

[7] Taleb, T., Kato, N., and Nemoto, Y.
An explicit and fair window adjustment method to enhance TCP efficiency and fairness over multiple satellite networks.
IEEE Journal on Selected Areas in Communications, **22**, 2 (Feb. 2004), 371–387.

[8] Marchese, M., Rossi, M., and Morabito, G.
PETRA: Performance enhancing transport architecture for satellite communications.
IEEE Journal on Selected Areas in Communications, **22**, 2 (Feb. 2004), 320–332.

[9] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, R., Scott, K., and Weiss, H.
Delay-tolerant networking: An approach to Interplanetary Internet.
IEEE Communications Magazine, **41**, 6 (June 2003), 128–136.

[10] Akyildiz, I. F., Akan, O. B., Chen, C., Fang, J., and Su, W.
InterPlanetary Internet: State-of-the-art and research challenges.
Computer Networks Journal, **43**, 2 (Oct. 2003), 75–113.

[11] Wang, R., Taleb, T., Jamalipour, A., and Sun, B.
Protocols for reliable data transport in space Internet.
IEEE Communications Surveys and Tutorials, **11**, 2 (Second Quarter 2009), 21–32.

[12] Bisio, I., Araniti, G., and De Sanctis, M.
State of the art and innovative communications and networking solutions for a reliable and efficient Interplanetary Internet.
International Journal on Advances in Internet Technology, **3**, 1 (Feb. 2010), 118–127.

[13] Caini, C., Cruickshank, H., Farrell, S., and Marchese, M.
Delay-and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications.
Proceedings of the IEEE (invited paper), **99**, 11 (July 2011), 1980–1997.

[14] Zhang, Z.
Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges.
IEEE Communications Survey and Tutorials, **8**, 1 (First Quarter 2006), 24–37.

[15] The Space Internetworking Strategy Group (SISG). Recommendations on a strategy for space internetworking, IOAG.T.RC.002.V1, Report of the Interagency Operations Advisory Group, NASA Headquarters, Washington, DC, Aug. 1, 2010.

[16] Scott, K., and Burleigh, S. Bundle protocol specification. IETF Request for Comments RFC 5050, Nov. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt.

[17] Demmer, M., and Ott, J. Delay tolerant networking TCP convergence layer protocol. IETF DTNRG IRTF Research Group, <draft-irtf-dtnrg-tcp-clayer-02.txt>, Nov. 2008. [Online]. Available: http://www.ietf.org/internet-drafts/draft-irtf-dtnrg-tcp-clayer-02.txt.

[18] Kruse, H., and Ostermann, S. UDP convergence layers for the DTN bundle and LTP protocols. IETF DTNRG IRTF Research Group, <draft-irtf-dtnrg-udp-clayer-00.txt> (Work in progress), Nov. 2008. [Online]. Available: http://tools.ietf.org/id/draft-irtf-dtnrg-udp-clayer-00.txt.

[19] Wood, L., McKim, J., Eddy, W. M., Ivancic, W., and Jackson, C. Using Saratoga with a bundle agent as a convergence layer for delay-tolerant networking. (Work in progress as an internet-draft, <draft-wood-dtnrg-saratoga-11>, Oct. 2012. [Online]. Available: https://tools.ietf.org/html/draft-wood-dtnrg-saratoga-11.

[20] Ramadas, M., Burleigh, S., and Farrell, S. Licklider Transmission Protocol–Specification. IETF Request for Comments RFC 5326, Sept. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5326.txt?number = 5326.

[21] Burleigh, S. Delay-tolerant networking LTP convergence layer (LTPCL) adapter. IETF Internet-Draft, <draft-burleigh-dtnrg-ltpcl-05>, Apr. 2013. [Online]. Available: http://tools.ietf.org/pdf/draft-burleigh-dtnrg-ltpcl-05.pdf.

[22] Caini, C., Cornice, P., Firrincieli, R., and Lacamera, D. A DTN approach to satellite communications. *IEEE Journal on Selected Areas in Communications* (special issue on delay and disruption tolerant wireless communication), **26**, 5 (June 2008), 820–827.

[23] Ramadas, M. Study of a protocol and a priority paradigm for deep space data communication. Ph.D. thesis, Russ College of Engineering and Technology, University of Ohio, Athens, June 2007.

[24] Wyatt, J., Burleigh, S., Jones, R., Torgerson, L., and Wissler, S. Disruption tolerant networking flight validation experiment on NASA's EPOXI. In *Proceedings of the2009 First International Conference on Advances in Satellite and Space Communications (SPACOMM)*, Colmar, France, July 2009, pp. 187–196.

[25] Wang, R., Gutha, B., and Rapet, P. K. Window-based and rate-based transmission control mechanisms over space-Internet links. *IEEE Transactions on Aerospace and Electronic Systems*, **44**, 1 (Jan. 2008), 157–170.

[26] Wang, R., and Horan, S. Protocol testing of SCPS-TP over NASA's ACTS asymmetric links. *IEEE Transactions on Aerospace and Electronic Systems*, **45**, 2 (Apr. 2009), 790–798.

[27] Wang, R., Shrestha, B., Wu, X., Wang, T., Ayyagari, A., Tade, E., Horan, S., and Hou, J. Unreliable CCSDS File Delivery Protocol (CFDP) over cislunar communication links. *IEEE Transactions on Aerospace and Electronic Systems*, **46**, 1 (Jan. 2010), 147–169.

[28] Wang, R., Burleigh, S., Parik, P., Lin, C. J., and Sun, B. Licklider Transmission Protocol (LTP)-based DTN for cislunar communications. *IEEE/ACM Transactions on Networking*, **19**, 2 (Apr. 2011), 359–368.

[29] Wang, R., Wu, X., Wang, T., Liu, X., and Zhou, L. TCP convergence layer-based operation of DTN for long-delay cislunar communications. *IEEE Systems Journal*, **4**, 3 (Sept. 2010), 385–395.

[30] Hu, J., Wang, R., Zhang, Q., Wei, Z., and Hou, J. Aggregation of DTN bundles for space Internetworking system. *IEEE Systems Journal*, **7**, 4 (Dec. 2013), 658–668.

[31] Wang, R., Wei, Z., Zhang, Q., and Hou, J. LTP aggregation of DTN bundles in space communications. *IEEE Transactions on Aerospace and Electronic Systems*, **49**, 3 (July 2013), 1677–1691.

[32] Bezirgiannidis, N., Burleigh, S., and Tsaoussidis, V. Delivery time estimation for space bundles. *IEEE Transactions on Aerospace and Electronic Systems*, **49**, 3 (July 2013), 1897–1910.

[33] Lee, D. C., and Baek, W. Expected file-delivery time of deferred NAK ARQ in CCSDS file-delivery protocol. *IEEE Transactions on Communications*, **52**, 8 (Aug. 2004), 1408–1416.

[34] Yu, Q., Sun, X., Wang, R., Zhang, Q., Hu, J., and Wei, Z. The effect of DTN custody transfer in deep-space communications. *IEEE Wireless Communications*, **20**, 5 (Oct. 2013), 169–176.

[35] Durst, R., Miller, G., and Travis, E. TCP extensions for space communication. *ACM/Kluwer WINET Journal*, **3**, 5 (Oct. 1997), 389–403.

[36] Bezirgiannidis, N., and 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 2010*, Moscow, Oct. 2010. [Online] http://www.spice-center.org/files/publications/icumt-2010.pdf.

[37] Burleigh, S. Interplanetary overlay network design and operation V3.2.1. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, JPL D-48259, July 2014. [Online] http://sourceforge.net/projects/ion-dtn/files/latest/download?source = files.

[38] Hogie, K., Criscuolo, E., and Parise, R. Using standard Internet protocols and applications in space. *Computer Networks Journal*, **47**, 5 (Apr. 2005), 603–650.

Qian Yu received her Bachelor of Engineering degree in electronic information science and technology from East China Jiao Tong University, China, in 2011 and her Master of Science degree in engineering from Soochow University, China, in 2014.

She is currently working toward her Ph.D. degree at Soochow University. Her research interests are in computer networks and communications systems.

Scott C. Burleigh is currently a principal engineer at Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. As a participant in CCSDS he was a co-author of the specifications for the CFDP and the AMS; while as a member of the DTN Research Group of the IRTF he co-authored the specifications for the DTN BP (RFC 5050) and the DTN LTP for delay-tolerant ARQ (RFC 5326). He has developed implementations of these protocols that are designed for integration into deep space mission flight software, with the long-term goal of enabling deployment of a delay-tolerant Interplanetary Internet.

Ruhai Wang (M’03) received the Ph.D. degree in electrical engineering from New Mexico State University, Las Cruces, NM, in 2001.

He is a Professor in the Phillip M. Drayer Department of Electrical Engineering at Lamar University, Beaumont, TX. He currently also serves as a Chair Professor at Soochow University, China, and a Principal Scientist at Nanjing University, China. His current research interests include space communications and networks.

Kanglian Zhao received his Bachelor of Engineering degree in electronic science and technology from Nanjing University, China, in 2003.

He is currently a lecturer in the School of Electronic Science and Engineering at Nanjing University, China. His research interests include wireless terrestrial communications, satellite communications, and networking.