Delivery Time Estimation for Space Bundles

Nikolaos Bezirgiannidis, *Member, IEEE*, Scott Burleigh, and Vassilis Tsaoussidis, *Senior Member, IEEE*

***Abstract*— We present a method for predicting delivery time of bundles in space internetworks. Bundle Delivery Time Estimation tool exploits Contact Graph Routing, predicts bundle route and calculates plausible arrival times along with the corresponding probabilities. Latency forecasts are performed in an administrative node with access to an instrumentation database appropriate for statistical processing. Through both analysis and experimentation, we demonstrate that estimates of bundle earliest plausible delivery time and destination arrival probabilities can be provided.**

***Index Terms*—Bit error rate, Disruption Tolerant Networking, Prediction methods, Routing protocols**

I. Introduction

Internetworking in Space has gained momentum recently, primarily for two main reasons. Firstly, Space Internetworking allows for better exploitation of resources, which in turn allows space engineers to communicate with deep space more easily and safely. Secondly, Space Internetworking designates a new space networking era where interoperability, interagency communication and unification of space and terrestrial networking are feasible.

In response to increased agency interest in internetworked space communications architectures, the Interagency Operations Advisory Group (IOAG) chartered a Space Internetworking Strategy Group (SISG) in 2007 "to reach international consensus on a recommended approach for transitioning the participating agencies towards a future 'network centric' era of space mission operations", which delivered the preliminary Operations Concept for a Solar System Internet (SSI) to the second Inter-Operability Plenary (IOP-2). The document [1] provided a top-level definition of SSI operations including a statement of requirements, which included two specific requirements that reflect necessary space-oriented restrictions:

> *TIMELINES: The network shall allow timely delivery, as required by the user, via managing the timing for delivery of the forward link product. Users will need to know the predicted epoch by which a given forward product will reach the destination node.*
>
> *PREDICTABILITY: It shall be possible to identify all provider components' latency and the resulting earliest/latest physical delivery times under normal conditions of SSI network operation.*

Practically, these objectives call for methods and protocols to estimate accurately and efficiently bundle delivery paths and delivery timeline. The present work is motivated precisely by those objectives: we introduce a technique for estimating bundle delivery time in a delay-tolerant network deployed in interplanetary space.

Coarse-grained (i.e., to the nearest second) estimation of the time of delivery of an application data item sent through the terrestrial Internet is trivial: barring a significant and unanticipated failure of infrastructure, transmitted data always arrive at their destination within a fraction of a second after transmission, usually a small fraction. Even in the event of transient data loss due to congestion, the end-to-end delivery of the transmitted data, including retransmission as necessary, consumes a few more milliseconds. Most Internet applications rely on this predictable and extremely low delivery latency in order to provide satisfactory communication service to users.

In the Solar System Internet, coarse-grained estimation of the time of delivery of an application data item will not be trivial. The distances between communicating entities may be very large, measured in tens to thousands of light seconds, and may vary widely (e.g., Saturn is about six times further from the Sun than Mars is). Moreover, those distances are constantly changing as solar system objects follow their different orbital paths. In addition, since only precisely directed radiation can enable communication over such vast distances (because simple broadcast radiation loses signal strength far too rapidly to be detectable by readily deployable instruments), the rotation of planetary bodies introduces additional latency by periodically making the reception of directed radiation impossible - interrupting communication altogether for some lengthy but predictable interval. Even when communication between entities in interplanetary space is geometrically possible it may in practice be temporarily impossible since transmission and/or reception equipment on one entity or another may be disabled, e.g., due to transient spacecraft operational constraints. In this context, such

Manuscript received May 15, 2012; revised December 4, 2012. The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013_FP7-REGPOT-2010-1, SP4 Capacities, Coordination and Support Actions) under grant agreement n° 264226 (project title: Space Internetworking Center-SPICE). This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein. Also, some of the research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

N. Bezirgiannidis is with the Space Internetworking Center, Democritus University of Thrace, Greece (e-mail: nbezirgi@ ee.duth.gr). S. Burleigh is with Jet Propulsion Laboratory, CA 91109 USA (e-mail: scott.c.burleigh@jpl.nasa.gov). V. Tsaoussidis is with the Space Internetworking Center, Democritus University of Thrace, Greece (e-mail: vtsaousi@ ee.duth.gr).

The original IEEE publication is available online:
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6558026

considerations render end-to-end data delivery latency not just lengthy but also somewhat challenging to predict: delay in conveying information about the communications state of an entity diminishes our confidence in the validity of this information when it finally arrives – and any failure of communication operations due to invalid information may impose yet more delay in communication.

Note that this already challenging problem statement presupposes successful communication whenever links are available. In reality the problem is much worse: data error rates are high in interplanetary communication by directed radiation, and any data loss may require retransmission. Retransmission may again require tens to thousands of seconds; and may itself fail and require further retransmission, resulting in unpredictable delivery latency.

By and large, a purely deterministic approach to predict bundle delivery latency in the SSI is rather scientifically naïve. Instead, an analysis of the likelihood for each bundle to follow some path, which incorporates both transmission latencies and retention latencies (contact interruption intervals) and considers the most plausible retransmission scenarios, allows for a weighted probabilistic delivery latency profile to be computed.

Our approach departs from this observation and introduces a novel method for estimating the bit error rate (BER) on each link. It uses recent network processing statistics to calculate the mean expected number of retransmissions on each segment of the end-to-end path and a binary search algorithm to estimate the expected BER. Network processing statistics will be supported by the DTN network management infrastructure. Further optimization of our current analysis procedures is certainly possible; however, we present here an important first step.

In Section II we present the necessary architectural background and we briefly discuss the relevant work on delay estimation in space. Section III includes a description of the core application functionality and of the algorithms used to predict future BERs and to estimate bundle delivery times. In Section IV we present a sample bundle transmission scenario and demonstrate our application's results on that. In Section V we discuss several open issues and finally we conclude in Section VI.

II. Background

A. Background

Delay-Tolerant Networking (DTN [2]) is a communications architecture that targets challenging environments where traditional networking protocols typically fail. Such extraordinary conditions are the extremely long signal propagation latencies and routine transient network partitions that characterize the Solar System Internet. In practice, both of these elements result in potentially large "round-trip" delay (the time required for data to reach its destination and a subsequent acknowledgment to reach the data's source), hence the name "Delay-Tolerant Networking". However, the prominence of transient network partitioning as a contributor to this delay has led to the use of the alternate formulation "Disruption-Tolerant Network" to refer to the same architecture.

A central element of DTN is an overlay network protocol called "Bundle Protocol" (BP [3]) that operates flexibly above either the transport or the network layer. The various protocols that may be operated immediately under BP in a DTN protocol stack are termed "convergence layer protocols". When DTN is used to interconnect networks based on the Internet architecture, the convergence layer protocol that is most frequently used is the familiar Transmission Control Protocol (TCP). In interplanetary networking, however, TCP will typically be set aside in favor of the Licklider Transmission Protocol (LTP [4]). Like TCP, LTP performs automatic detection of data loss and automatic retransmission of lost data, but unlike TCP it does not exhibit degraded performance over extremely long round-trip delays.

The frequent changes in link availability among entities in an interplanetary network result in a constantly changing network topology. The nature of interplanetary communications makes route computation through this highly dynamic topology really challenging. On the one hand, the very long signal propagation delays of the interplanetary medium make the use of Internet-like routing protocols to announce these changes in topology impractical. On the other hand, space mission operators long ago realized that spacecraft could only communicate by anticipating transmission opportunities long before they occurred; the carefully planned and scheduled contact intervals computed on Earth and described in command sequences transmitted to spacecraft days or weeks before they occur may be used by BP route computation procedures as an authoritative forecast of topology changes that could not be discovered in time to be considered in routing.

Currently the only BP route computation mechanism that is based on this principle is "Contact Graph Routing" (CGR [5]). CGR operates by using the operator-provided contact plan to construct a directed graph from the local DTN node to the destination node of a given bundle, where each vertex of the graph is one of the planned contacts. By finding the path through this graph that results in the earliest time of delivery of the bundle at the destination node, CGR determines which of the local node's neighboring nodes the bundle should be passed to for further forwarding.

At the present time, the only implementation of the DTN protocols that includes an implementation of CGR is Interplanetary Overlay Network (ION [6]), developed at the Jet Propulsion Laboratory, California Institute of Technology. As no other DTN implementation is currently configurable for successful operation in the Solar System Internet, we have used ION in all of the Bundle Delivery Time Estimation (BDTE) research we have performed to date.

B. Related Work

The majority of the published works [7-12] that study file transmission times in space environments examine CCSDS File Delivery Protocol (CFDP [13]) and its different modes. Bundle transmissions that use BP as an overlay layer have not yet been studied in this context. BDTE, to the best of our knowledge, is the first attempt to approach this subject without regard to any specific applications or transmission modes and without limiting itself to file transfers.

Lee and Baek in [7-8] have studied different CFDP schemes and delivery time expectations in deep-space scenarios. In [7], CFDP Deferred NAK mode, with functionality equivalent to LTP, is examined, while in [8] the CFDP Immediate NAK mode is under evaluation. In both papers, the authors consider single-hop file transmissions in a Mars-to-Earth scenario and define rules for computing retransmission timeout intervals that minimize expected file-delivery time with the constraint that throughput efficiency is not compromised. They evaluate variation in expected file-delivery time with varying BER, PDU length, and file size, and provide both analytic and simulation results.

Ka-band channels and their weather dependencies are studied in [9], where the authors model the effect of weather on Ka transmission as a Gilbert-Elliot channel with two weather states ("good" and "bad") and analytically calculate file delivery latency with probabilities that depend on the channel weather on each transmission round. The only latency component considered here is propagation delay and one of the measured metrics is the average number of transmission rounds required to complete the file delivery (termed *spurts*).

One of the main differences between the three aforementioned analytic studies and our technique is the fact that they calculate the expected number of transmission rounds, which is a single number that corresponds to a precise arrival time at destination. We consider the same metric but use it only as an intermediate metric that helps estimate BER and ultimately provide the distinct probabilities for each possible number of rounds that the transmission may last.

In [10] the authors study the performance of CFDP Deferred NAK mode for Mars-to-Earth communications over Ka-band channels, evaluating file transfers over a deep-space link in terms of latency and storage requirements. The authors in this paper analyze the probability distribution for different number of transmission rounds that file delivery will last, and provide both analytical and simulation results for file transmissions of various sizes, with different error rates and "Data Completeness Requirement" percentages. The analytical types they use to extract transmission round probabilities are similar to those used in our method. However, transmission delay is ignored as insignificant and thus the results are not sensitive to data rates. As in the other studies noted above, only single-hop transmission paths are examined.

Furthermore, various studies have been presented about the performance of transport protocols or protocol stack architectures in different space environments, including satellite [11, 15], cislunar [12, 14] and deep space [16-17] communications. The majority use file or PDU delivery time as a metric to evaluate the performance of the proposed configurations.

A multi-hop, heterogeneous satellite scenario that includes LEO and GEO satellites is studied in [11], where data file transfers are evaluated under different protocol stack approaches. In [15], file transmission times, transmission patterns and throughput are also studied as a means to compare the performance of different window-based and rate-based control mechanisms in satellite communications.

Cislunar communications and the varying conditions that characterize them are studied in [12, 14]. In [12], unreliable CFDP mode and its transmission effectiveness are evaluated, while in [14] the authors present an evaluation comparison of different DTN convergence layers and their performance in terms of goodput.

Within the same context, deep space communications are studied in terms of protocol performance through emulation measurements on the delivery latency [16-17]. In [16], the impacts of LTP segment size, LTP block size, and bundle size are studied in a Mars-to-Earth file transmission scenario. In [17], an erasure-coding scheme is compared with typical ARQ functionality in terms of file deliveries in space. The tradeoff between the gain in delivery latency and the loss in redundancy introduced by erasure coding is studied, while the metrics used are normalized to the Bandwidth-Delay Product.

The present work is, to the best of our knowledge, the first method that is able to estimate bundle delivery times in deep space network environments, considering the entire network contact plan and computing estimated results for multi-hop bundle deliveries and transmissions that might include alternate routes. BDTE also attempts to predict channel error rates based on link history observations, in contrast to all the aforementioned works, which use the whole spectrum of possible error rates as input. We believe BDTE therefore can provide accurate and robust results in forecasting bundle delivery times.

III. Application Description

BDTE is in essence an administrative network simulation tool that applies the CGR algorithm on every network node throughout the route of the bundle. The BDTE algorithm performs hop-by-hop simulations, provides possible arrival times for each hop, and continues iteratively through the entire predicted bundle route, ultimately resulting in the arrival time at the final destination. The calculated latency for each hop is based on deterministic and stochastic latency components. The former comprises propagation delay (also referred to as One-Way-Light-Time or OWLT) and transmission delay for bundle delivery (including overhead) via the link channel, i.e., the length of time that will be required simply to transmit the bundle given the maximum transmission rate on the link. The stochastic component is introduced by uncorrected channel errors, which compel packet retransmissions; it accounts for the propagation and transmission delay for retransmitted packets.

In our analysis we make some simplifying assumptions. We assume that processing delays are insignificant in contrast to the long propagation delays. We also omit consideration of queuing latency, assuming the bundle is transmitted at the highest possible priority. This is not always the case in bundle transmissions; however, consideration of this factor would introduce complexity into our analysis that is out of the scope of this paper. This is further discussed in section V.

BDTE's computation is based on the fact that the deterministic components of the bundle's latency can be accurately calculated, whereas the stochastic latency can only be statistically predicted using each link's history observations. The result of this analysis is a link error rate forecast that provides several estimates of the number of transmission rounds that may be required for successful bundle delivery to the next node, each with its own probability. For each possible number of transmission rounds, a different delivery time to the next node is calculated. This time is then used as the transmission initiation time for the next hop of the route and a new simulation is then performed. This method continues consecutively to the final destination and ultimately results in a set of distinct bundle arrival times, with different probabilities that theoretically sum up to 100%.

This network simulation and all analyses are performed in an administrative node that may or may not be a part of the space internetwork. This node is assumed to have current knowledge of the overall network contact plan, as well as access to a central database that contains network instrumentation statistics through time. Past measurements from this statistics database are used to predict channel error rates for future bundle transmissions. The accuracy of BDTE will always be limited to the accuracy of these information resources.

We note that BER observations are performed in the convergence layer and thus incorporate the uncorrected errors that have eluded channel Forward Error Correction (FEC).

In the following subsections we describe:

a) The main BDTE functionality including Application input, output, and the core delivery time estimation algorithm,

b) The statistics database, some of its useful fields and how they are used to calculate the *Average Transmission Rounds* (*ATR*) metric for past time intervals,

c) The equations that are used to calculate (for past time intervals) BER, enabling computation of packet error rate (PER), thereby enabling calculation of distinct probabilities for each number of transmission rounds, which in turn enables computation of *ATR* together with the binary search algorithm that is used to estimate the inverse, i.e., BER from *ATR*,

d) The statistical forecasting method that constructs a time series from past BER observations and uses it to predict BER for the time of bundle transmission initiation, and

e) The assumptions that we have applied in our method.

TABLE I
NOTATION

Symbol	Quantity
OWLT	One Way Light Time (Propagation Delay)
P{p.r. ><= k}	Probability that the number of packet transmission rounds is bigger than / smaller than / equal to *k*
P{t.r. ><= k}	Probability that the number of bundle transmission rounds is bigger than / smaller than / equal to *k*
BDT	Bundle Delivery Time
TPList	Time-Probability List: a list of BDTs at destination and the corresponding probabilities
MBS	Mean Block Size
MPL	Mean Packet Length
ANP	Average Number of Packets (per bundle)

a) Main BDTE functionality

Application input:
- *sending_endpoint* (node that initiates bundle transmission)
- *destination_endpoint*
- *Bundle_creation_time*
- Bundle's *lifetime*
- Bundle size
- Convergence protocol packet size

Application output:
List of bundle delivery times and corresponding probabilities.

TABLE II
BDTE ALGORITHM

```
// Initialization
current_node = sending_endpoint;
end_node = bundle destination_node;
initial_probability = 1;
start_time = Bundle_creation_time;
bundle_expiration_time = start_time +lifetime;

CalculateNextHopDeliveryTimes (current_node, end_node,
initial_probability, start_time)
{
CGR algorithm (current_node, destination_node, start_time) →
next_node, start_xmit_time;
Extract BER statistics for span [current_node - next_node] from network
statistics database;
Construct time series for BER through time, for span [current_node -
next_node];
Predict BER from constructed time series at start_xmit_time;
Calculate PER from BER, using (5);
for (k =1..max_transmission_rounds)
{
    Calculate P{t.r. = k} using (13);
    if (P{t.r. = k} ≤ probability_threshold)
        break;
} // k has now been set as the max number of transmission rounds to be
examined

for (j=1..k transmission rounds)
{
```

$Propagation\ Delay = OWLT \cdot (2j-1)$;

$Transmission\ Delay = bundle\ size \cdot \left(1+\sum_{i=1}^{j-1} PER^{i}\right) \Big/ channel\ rate$;

$Total\ Delay = Propagation\ Delay + Transmission\ Delay$;

$BDT_j = start\_xmit\_time + Total\ Delay$;

$P_j = initial\_probability \cdot P\{t.r. = j\}$

```
if (next_node == end_node)
{
Store pair [BDT_j – P_j] in final TPList;
continue;
}
current_node = next_node;
start_time = BDT_j;
initial_probability = P_j;
CalculateNextHopDeliveryTimes(current_node, end_node,
initial_probability, start_time);
} // for (j=1..k transmission rounds)
} // CalculateNextHopDeliveryTimes
```

In Table II we present the core BDTE algorithm that performs the simulations and leads to possible bundle delivery times at destination. This is a recursive algorithm that concludes when *next_node* is the *destination_node*, i.e., when last hop probabilities have been calculated. Its output is a list of pairs, namely *TPList*. Each pair consists of a bundle delivery time at the receiving node and the corresponding probability. The reason for using the two configuration parameters *max_transmission_rounds* and *probability_threshold* is to control the number of iterations and reduce the computational cost. Since the number of transmission rounds could theoretically be infinite with probability that tends to zero, we can either set a maximum number of transmission rounds or a probability threshold below which the calculation is negligible. In our testing configuration we have used both of these control parameters and have set *max_transmission_rounds* = 4 and *probability_threshold* = 0.001.

b) Statistics Database and ATR calculation

ION instrumentation provides information for the DTN network management procedures that are currently being designed. Each DTN node that uses ION will keep records of several types of events in both bundle and convergence layers and will measure incoming as well as outgoing bytes, bundles, and convergence layer packets. For the purposes of BDTE we assume that these measurements will be transmitted to a central administrative node and stored in an instrumentation database for detecting network defects and failures and for further processing.

Some of the measurements stored in this central DB can be used to evaluate the quality of a given link that is a part of the space internetwork and to predict its behavior in the future. In order to quantify the link quality, we consider a metric called *Average Transmission Rounds* (*ATR*) that indicates the anticipated number of retransmission rounds for the convergence layer.

In equation (1) we consider the specific LTP export span in order to calculate the average number of retransmission rounds required to deliver each bundle successfully to the next hop and we thus calculate *ATR*. The same results could be extracted if the corresponding LTP import span was used.

$$\begin{aligned} &ATR = \\ &1 + \textit{Average LTP retransmission rounds} = \\ &1 + \frac{\textit{NAK reports rcvd} + \textit{checkpoints rexmitted}}{\textit{xmit sessions completed} + \textit{xmit session cancelled}} \end{aligned} \tag{1}$$

$$MBS = \frac{\textit{Data segments dequeued}}{\textit{xmit sessions completed} + \textit{xmit sessions cancelled}} \tag{2}$$

$$MPL = \frac{\textit{Data bytes dequeued}}{\textit{Data segments dequeued}} \tag{3}$$

$$ANP = round\left(\frac{\textit{Mean block size}}{\textit{Mean packet length}}\right) \tag{4}$$

With (1) we can calculate *ATR* for time intervals stored in the DB. Using *ATR* along with the useful metrics *MBS*, *MPL*, and *ANP* extracted from equations (2), (3) and (4), we can estimate the average observed BER with a technique that is introduced in the next subsection.

c) Calculating BER using ATR

Our algorithm uses the distinct probabilities for each number of transmission rounds that bundle delivery may require. The already computed *ATR* is the average value of transmission rounds and includes no such information for the distinct number of rounds. A solution for this would be achieved if BER could be calculated using *ATR*. However, there can be no such straightforward calculation; therefore, we present in this subsection an inverse method that uses the explicit calculation of *ATR* from BER and applies a binary search algorithm to estimate *BER* with adequate precision.

In our analysis, we assume that bit errors are independent. If *s* is the packet length in bits, and assuming that all packets are of equal length, the loss probability of each packet is termed Packet Error Rate (PER) and can be calculated as follows:

$$PER = 1 - (1 - BER)^s \text{ , where } s = 8 \cdot MPL \tag{5}$$

The probability that a packet reaches the next node on the end-to-end path at exactly one transmission round (more precisely, less than two rounds) is:

$$P\{\text{p.r.} = 1\} = P\{\text{p.r.} < 2\} = 1 - PER \tag{6}$$

Bundle sizes ordinarily exceed convergence layer packet size. So, when BP delivers a bundle to the convergence layer beneath, the bundle is normally truncated into multiple segments to be delivered to the link layer. Since we have already assumed independent bit errors, the probability that a given packet is successfully transferred is independent from the transfer success probability of all other packets. Therefore, if a bundle consists of *n* packets, the probability $P\{t.r. = 1\}$ that its transmission lasts exactly one round corresponds to the probability that all *n* packets are successfully transmitted at the first transmission round, which equals to the product of the probabilities $P\{p.r. = 1\}$, for all *n* bundle packets:

$$P\{\text{t.r.} = 1\} = P\{\text{t.r.} < 2\} = \left(1 - PER\right)^n \tag{7}$$

In our algorithm, *n* is equal to *MBS*, which is calculated using (2). In order to calculate the probabilities for transmission rounds greater than 1, we have to consider the convergence protocol functionality. In our example, we use LTP, which includes ARQ-based retransmissions. Requests for retransmissions are initiated upon the delivery of the last packet of the block, namely the LTP segment flagged as End Of Block (EOB). In case of an unsuccessful EOB delivery, no positive or negative acknowledgment report is sent from the destination node. Hence, the retransmission timer at the sender expires and triggers EOB retransmission, thus delaying the block delivery by one round. The exact amount of delay produced is equal to the timeout time length plus the OWLT required to retransmit the EOB. In [6] timeout is computed as twice the OWLT plus the imputed inbound and outbound queuing delays in both communicating nodes, for the enqueuing and dequeuing of EOB and the report segment. As noted above, however, we assume that these queuing delays are insignificant in contrast to the huge space propagation delays. Consequently, timeout equals to 2•OWLT, or in other words, the additional delay is simply one transmission round. Hence the time granularity used in our analysis is in terms of transmission rounds.

An LTP block, which we here assume corresponds into a single bundle, is truncated into n segments, n-1 regular red-block segments and a last one, denoted as EOB. In the same way as before, the probability that a packet is transmitted in less than k rounds equals to 1 minus the probability that the packet is not successfully transferred in all k-1 first rounds:

$$P\{\text{p.r.} < k\} = 1 - PER^{k-1} \tag{8}$$

The corresponding independent probability for n-1 packets to be transferred in less than k rounds is:

$$P\{\text{t.r.} < k\} = \left(1 - PER^{k-1}\right)^{n-1} \tag{9}$$

Using (9) we calculate the probability for the first n-1 bundle packets to be successfully delivered in exactly k transmission rounds, $k \geq 1$:

$$\begin{aligned} &P\{\text{t.r.} = k,\ \text{for } n-1 \text{ red-block segments}\} = P\{\text{t.r.} < k+1\} - P\{\text{t.r.} < k\} = \\ &= \left(1 - PER^{k}\right)^{n-1} - \left(1 - PER^{k-1}\right)^{n-1} \end{aligned} \tag{10}$$

which is always greater or equal to zero, since

$$\begin{aligned} &PER \leq 1 \Rightarrow \\ &PER^{k} \leq PER^{k-1} \Rightarrow \\ &1 - PER^{k-1} \leq 1 - PER^{k} \Rightarrow \\ &\left(1 - PER^{k-1}\right)^{n-1} \leq \left(1 - PER^{k}\right)^{n-1} \Rightarrow \\ &\left(1 - PER^{k}\right)^{n-1} - \left(1 - PER^{k-1}\right)^{n-1} \geq 0 \end{aligned}$$

We have thus calculated the probability for the transmission rounds of the n-1 first LTP segments. In order to estimate the total transmission time of the LTP block, we have to consider its checkpoint-based ARQ functionality, assuming that LTP configuration incorporates one checkpoint per block, the EOB segment. In case EOB transfer fails, the destination node transmits no report segment, resulting in timer expiry and EOB retransmission. This event triggers extra transmission rounds until the successful delivery of EOB.

We denote the probability of m number of lost EOBs during a bundle transmission as $P\{m \text{ lost EOBs}\}$. For a bundle transmission that lasts k rounds, the probability that m EOB packets are lost in the first k-1 rounds is given by the probability mass function of the binomial distribution:

$$P\{m \text{ lost EOBs in } k\text{-1 rounds}\} = \binom{k-1}{m} \cdot PER^{m} \cdot (1 - PER)^{k-1-m}$$

The last EOB in the k-th round arrives successfully with probability 1-PER, to complete bundle reception after exactly k rounds. Thus, the total probability that exactly m EOBs are erroneously transferred in the first k-1 transmission rounds and the k-th EOB is successfully transferred is:

$$P\{m \text{ lost EOBs}\} = \binom{k-1}{m} \cdot PER^{m} \cdot (1 - PER)^{k-m} \tag{11}$$

If in a bundle transmission that lasts k rounds the EOB is retransmitted in m rounds ($m \leq k$), the other red-data segments of the block are not transmitted during these rounds. Therefore, the successful delivery of the n-1 segments has to be achieved within the remaining k-m rounds, with a probability computed with (10). Hence, there are k distinct cases for a bundle transmission that lasts exactly k rounds: the loss of 0, 1, 2,..., k-1 EOBs. The total probability of a successful bundle delivery in exactly k transmission rounds is the sum:

$$\begin{aligned} &P\{\text{t.r.} = k \mid \text{error-free backward channel}\} = \\ &\sum_{m=0}^{k-1} P\{m \text{ lost EOBs}\} \cdot P\{\text{t.r.} = k - m,\ \text{for } n-1 \text{ red-block segments}\} = \\ &\sum_{m=0}^{k-1} \binom{k-1}{m} \cdot PER^{m} \cdot (1 - PER)^{k-m} \cdot \left[\left(1 - PER^{k-m}\right)^{n-1} - \left(1 - PER^{k-m-1}\right)^{n-1}\right] \end{aligned} \tag{12}$$

for $k > 1$ and $m < k$.

In the degenerate case where a bundle is incorporated into a single LTP segment, which is also the EOB segment, the probability of k transmission rounds equals to:

$$P\{k-1 \text{ lost EOBs}\} = PER^{k-1}$$

We have so far assumed that the return (acknowledgment) channel is error-free. The plausibility of this assumption may be increased by the use of small Report Segments (RSs), or by the use of strong encoding schemes at the underlying link and/or physical layers, that greatly reduce the statistical significance of a RS loss. Nevertheless, the database field *checkpoints retransmitted* in (1) includes the lost RSs as well. So, for a more accurate result, RS error rate could be incorporated in (12) by adding the loss probability of j RSs, with $0 \leq j < k$ and $j + m < k$. Useful RSs (*i.e.*, not retransmissions of already received RSs) are not transmitted when an EOB has not arrived successfully. Thus, RSs are not transmitted in the m rounds in which EOB is lost, but only in k-m rounds. The successful or unsuccessful transmission of the k-th RS (i.e., the one transmitted after the k-th round) is not considered, since the transmission has been completed at k rounds. Therefore, the considered values for j are [0..k-m-1], the number of different combinations for j is $\binom{k-m-1}{j}$ and the corresponding probability for each j is

$$P\{j \text{ lost RSs in } k-m-1 \text{ rounds}\} = \binom{k-m-1}{j} \cdot PER_{RS}{}^{j} \cdot \left(1-PER_{RS}\right)^{k-m-1-j}$$

The final probability would become:

$$P\{\text{t.r.}=k\} = \sum_{j=0}^{k-m-1} \sum_{m=0}^{k-1} \left\{ \begin{array}{l} P\{j \text{ lost RSs in } k-m-1 \text{ rounds}\} \cdot P\{m \text{ lost EOBs}\} \cdot \\ P\{\text{t.r.}=k-m-j, \text{ for } n-1 \text{ red segments}\} \end{array} \right\} \Rightarrow$$

$$P\{\text{t.r.}=k\} = \sum_{j=0}^{k-m-1} \sum_{m=0}^{k-1} \left\{ \begin{array}{l} \binom{k-m-1}{j} \cdot PER_{RS}{}^{j} \cdot \left(1-PER_{RS}\right)^{k-m-1-j} \binom{k-1}{m} \cdot PER^{m} \cdot \\ \left(1-PER\right)^{k-m} \cdot \left[\left(1-PER^{k-m-j}\right)^{n-1} - \left(1-PER^{k-m-j-1}\right)^{n-1}\right] \end{array} \right\} \quad (13)$$

where PER_{RS} is the loss rate for the report segments, calculated from (5), if we assume the same BER in the return channel.

Theoretically the number k of transmission rounds could be infinite, with probability that tends to zero. However, as we have already mentioned, we apply the *max_transmission_rounds* and *probability_threshold* filters, in order to reduce the calculation cost for insignificant probabilities. Since we have so far calculated a finite number of significant probabilities for the distinct number of transmission rounds, we can use them to evaluate *ATR*:

$$ATR = \sum_{k} \left(k \cdot P\{\text{t.r.}=k\}\right) \quad (14)$$

In equations (5)-(14) we have presented a method that uses *BER* to compute the probabilities for a bundle to be successfully transmitted in 1, 2, ..., k rounds, as well as the aggregated *ATR*. This method has a two-fold significance: If BER is a known (or estimated) quantity, we can extract useful information about the bundle transmission in terms of delivery times at destination. On the other hand, if *ATR* is known, as with past DB measurements from (1), this method can be used in an inverse binary search algorithm (see Table III), which attempts to approximate PER from the known *ATR*. Channel BER can then be inferred, using the inverse of (5). This inverse calculation has a unique solution because *ATR* is a genuinely ascending function of PER.

The PER estimation algorithm parameters that need to be configured are the starting minimum and maximum error rates and the number of iteration steps. Similarly, an error threshold could also be applied to lead to a desired accuracy. However, this is a matter of application configuration and should be user-defined, based on the desired results and the computational resources. In our sample configuration we have used BER minimum = 10^{-8}, BER maximum = 10^{-5}, and iteration steps = 12.

TABLE III

PER ESTIMATION ALGORITHM

```
Input: ATR
Output: PER
Configuration parameters:
times = max number of iterations;
min = calcPerFromBer(10^-8);
max = calcPerFromBer(10^-5);

while ((times >= 0) && (min < max))
{
    times--;
    per = (min + max) / 2;
    ATRtemp = calcATR(per);
    if (ATRtemp < ATR)
    {
        min = per;
    }
    else
    {
        max = per;
    }
}
return (min + max) / 2;
```

d) Forecasting Method

In the previous subsections we have described a method to extract useful information from the instrumentation DB for past time intervals, which could be quantified as a BER value. This method can thus provide a time series that consists of BER values through time, for the links that form the predicted bundle route.

The composed time series can provide a means to predict the future link behavior, in terms of error rates. Our rationale for this assertion is that BER through time is not a random variable. It is generated by time-dependent events such as space weather and solar activities and, therefore, it is an auto-correlated variable and thus can be estimated using observation history.

Many different forecasting techniques have been developed during the last decades, from simple Moving Average to Exponential Smoothing method [18], ARMA and ARIMA [19]. The majority of them are developed to fit explicit models and thus apply better in specific time series. As a rule, in order to apply the most suitable forecasting model in a time series, one needs to study its evolution through time, as well as to analyze its trend and periodicity, if such exist. After this careful study, time series may need decomposition into components (e.g., periodic, trend), which can be used for forecasting.

BDTE, however, is designed to quickly and automatically reply to user input, with no intermediate manual inspection of any time series. Furthermore, the use of a specific prediction technique requires a complete and thorough analysis of error rates in space in general, for all times and seasons and for different weather and environmental conditions. The DTN statistics database, however, is still under design and therefore there is no access to sufficient space DB sampling and measurements; such an analysis is therefore out of the scope of this paper. For these reasons we cannot base our model on the actual space-channel behavior and form a realistic time series. Instead, we have used an exponential smoothing method as an initial forecasting method for testing purposes. It can be described as a simple and robust generic technique of time series forecasting that may fit different time series models. The rationale behind our choice is mainly the accuracy that can be obtained with exponential smoothing, with minimal effort in model identification [18]. The prediction method that BDTE uses may be optimized or reconsidered when access to real space measurements is possible and error rate distributions are studied in depth.

The triple exponential smoothing technique used is also referred to as Holt-Winters method, [20-21]. This is an extension of exponential smoothing model designed for time series with trends and seasonality and does not require a large amount of time series data. Although we have as yet had no access to real channel information, we can foresee that both trend and seasonality should be included in data analysis. The incorporation of trends can be justified by the fact that space channel behavior is greatly affected by space weather conditions, which may have a linear behavior in short time intervals. Seasonality analysis on the other hand is included due to the periodicity of planetary and space assets' movement. In a case of a BER time series where either trend or seasonal components (or both) are missing, the corresponding factor can be set to zero and thus excluded from the model construction.

We now briefly describe the Holt-Winters prediction technique we have used in our application to forecast BER values. The additive Holt-Winters prediction function (for time series with period length p) [notation from 20] has linear trend and additive seasonality. The time series rate is obtained from:

$$\overline{S}_t = a(S_t + P_t) + (1-a)(\overline{S}_{t-1} + r_t)$$

where S_t is the observed time series value at time t, P_t is the periodic adjustment increment for the t-th period, r_t is the trend adjustment increment for the t-th period, and a is a constant that determines how fast the exponential weights decline over the past periods.

The periodic and trend adjustment increments are calculated correspondingly as follows:

$$P_t = b(\overline{S}_t - S_t) + (1-b)P_{t-N}$$

$$r_t = c(\overline{S}_t - \overline{S}_{t-1}) + (1-c)r_{t-1}$$

with b and c the exponential weight constants for the periodic and trend components correspondingly. Time series forecasts T periods in the future are estimated as:

$$ES_{t+T} = \overline{S}_t + r_t T - P_{t+T-N},\ T = 1, 2, \ldots, N$$

The optimal values of a, b, and/or c are estimated by minimizing the squared one-step prediction error.

All statistical functions, including Holt-Winters model, the forecasting method, as well as the optimization technique, were integrated into ION from R, a free software environment for statistical computing [22].

In order to determine the periodicity of a specific time series, we follow an algorithm introduced in [23]. In this study, according to Peter Turchin, a way of determining the seasonal component of a time series is based on its Auto-Correlation Function (ACF). The statistical significance of ACF can be found with the use of a simple algorithm described in Table IV.

TABLE IV
ALGORITHM FOR STATISTICAL SIGNIFICANCE OF ACF

```
Calculate ACF for time series
Find first local maximum at lag T
if (ACF[T] >2 / sqrt(n))
{
    // we have "strong evidence of statistical periodicity"
    period =ACF[T];
}
else if (ACF[T/2]< -2 / sqrt(n))
{
    // we have "weak evidence of statistical periodicity"
    period =ACF[T];
}
else
{
    // Both are close to zero, so there is no evidence of periodicity.
    period = 1;
}
```

We note that the maximum period that may be available for this algorithm is half the sample size. The seasonal term is ignored if the BER time series has no periodic component. According to [21], for seasonal models, S, P and r initial values are inferred by performing a simple decomposition in the trend and seasonal components using moving averages on a number of initial periods, while a simple linear regression on the trend component is used for starting level and trend. For trend models without any seasonal component, start values for S and r are $S[2]$ and $S[2]$ - $S[1]$, respectively. For ordinary exponential smoothing, i.e., for BER time series with no seasonal and trend components, the start value for S is $S[1]$.

e) Model Assumptions

An important factor for accurate BER prediction is DB sampling, which we assume that is ideally performed for all spans in equal time intervals and not in random times. We also assume that LTP span configuration information exists in the database for all time intervals in the past. In practice, there could be several missing values, due to network unavailability or other reasons that may lead to DB sampling failure. In such cases interpolation techniques should apply.

For (13) we have assumed the same BER for both forward and backward channel, in order to compute PER_{RS}. The inverse binary search algorithm is otherwise unable to compute the 2 different BERs (forward and backward) and another, more sophisticated inverse search algorithm has to be applied, thus increasing the complexity of the application.

In our forecasting technique, we predict BER for the time interval containing the moment of bundle transmission initiation, rather than for the total bundle transmission interval. In other words, if bundle transmission time exceeds the time interval that has been predicted, BER prediction deviation from real value might exceed a small statistical error. We can state however that the statistical significance of this deviation decreases, since the majority of errors occur in the first transmission round of the bundle.

The equation that uses BER to calculate PER assumes independent bit errors (a Gaussian bit error distribution on the channel), which is not always the case; burst errors, for example, occur on space channels. On the other hand, the rate of snapshot capture for the statistics database is not expected to be high. This results in BER calculation over relatively long time intervals, which may be expected to exhibit “average” channel behavior. That said, burst errors are in essence outliers that are intentionally excluded from our prediction calculations.

For simplicity reasons, in PER calculation (5) we round the number of packets up to the next integer and consider them all equal to the input packet length. This assumption could lead to a significant error for small bundles that are truncated into LTP segments with a small last segment. For example if convergence layer packet length is 1400 bytes and the delivery time of a 1500-byte bundle is to be estimated, application will consider the transmission of a bundle that is truncated into two 1400-byte packets.

Finally, metrics measured during the DB time interval may have a significant standard deviation and, therefore, average values could be inaccurate metrics for block size (2), packet length (3) and number of packets in block (4).

IV. Sample Scenario

We provide a sample scenario of a bundle transmission from node 1 to node 3 via node 2 (see Fig. 1). Scenario parameters are available in Table V. Error rates between nodes 1 and 2 follow the time series depicted in fig 2a and include seasonality with period = 9 time slots, a linear trend, and a random error with normal distribution.

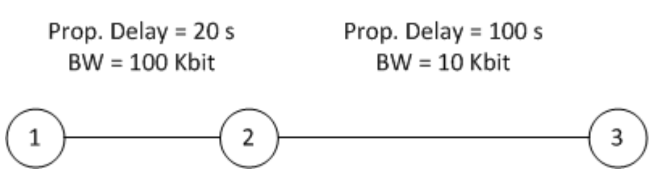
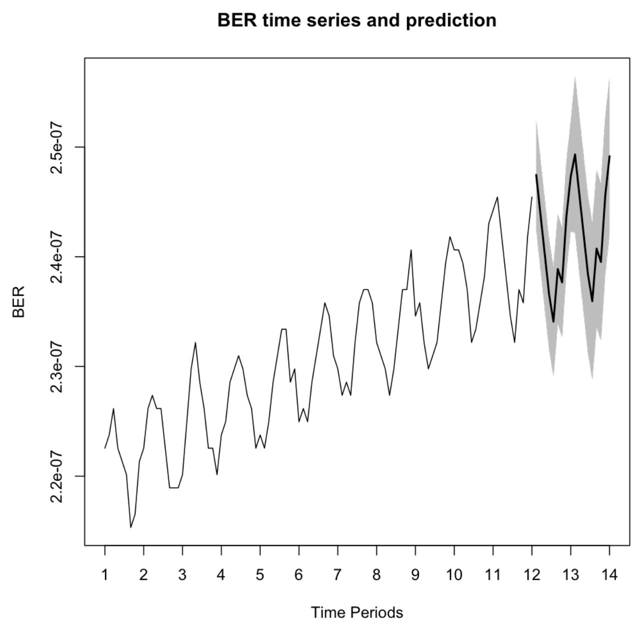


Fig. 1. Scenario topology.

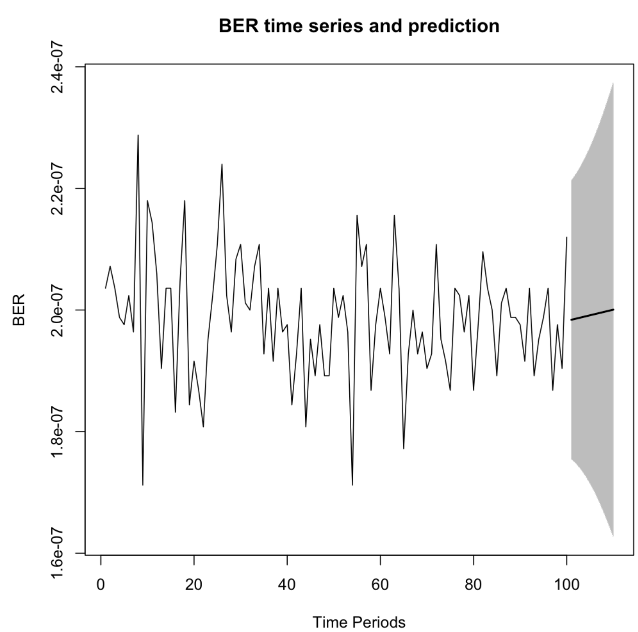
The predicted values for several periods of time are depicted in fig 2a, with the corresponding 95% confidence intervals. In the link between nodes 2 and 3 we have applied a random error distribution that BDTE interprets as a Holt-Winters model without seasonal component and with an insignificant trend as observed in Fig. 2b. The 95% confidence intervals can also be observed in figure 2.

At 11:00:00 a bundle transmission from node 1 to 2 is initiated. The predicted BER for this moment is $2.4177 \cdot 10^{-7}$ and the expected transmission rounds are calculated and displayed in table VI.

After the first hop calculations, 3 distinct cases are extracted for the bundle to arrive at node 2: after 21, 61, or 101 seconds (1, 2, or 3 transmission rounds) with probabilities 0.822864, 0.176609 and 0.000525 correspondingly. Each one of them is then treated separately in new simulations for the next hop (2-3) with transmission initiation time equal to the bundle arrival at node 2 or the contact opening time between 2 and 3, whichever of the two times is later. In our example, the contact between 2 and 3 is always on, so the transmission



a. BER time series with seasonality and trend



b. BER time series with random values

Figure 2. Observed BER and future values prediction. The shaded area corresponds to the 95% confidence intervals of predicted values.

initiation time from node 2 to 3 is 11:00:21, 11:01:01 and 11:01:41 correspondingly for the 3 distinct cases. The newly derived probabilities are multiplied with the initial ones to calculate the final probability for each delivery time at final destination.

The times at destination are then sorted and the cumulative probabilities are calculated, showing the probability that the bundle will have been delivered at the final destination node before a specific future time (see Table VII).

TABLE V
SCENARIO PARAMETERS

Parameter	Value
Packet size	1400 Bytes
Bundle size	100,000 Bytes
Propagation delay 1-2	20 s
Propagation delay 2-3	100 s
Bandwidth 1-2	100 Kbit/s
Bandwidth 2-3	10 Kbit/s
Transmission delay 1-2	1 s
Transmission delay 2-3	10 s

TABLE VI
BDTE CALCULATIONS FOR SCENARIO

Link	Bundle Xmit Time	Predicted BER	Initial Probability	Probability	Time needed (s)	Time at destination
1-2	11:00:00	$2.42 \bullet 10^{-7}$	1 1 1	0.82286 0.17661 0.00053	21 61 101	11:00:21 11:01:01 11:01:41
2-3	11:00:21	$2.02 \bullet 10^{-7}$	0.82286 0.82286 0.82286	0.84926 0.15037 0.00037	110 310 510	11:02:11 11:05:31 11:08:51
2-3	11:01:01	$2.02 \bullet 10^{-7}$	0.17661 0.17661 0.17661	0.84926 0.15037 0.00037	110 310 510	11:02:51 11:06:11 11:09:31
2-3	11:01:41	$2.03 \bullet 10^{-7}$	0.00053 0.00053 0.00053	0.84926 0.15037 0.00037	110 310 510	11:03:31 11:06:51 11:10:11

TABLE VII
CUMULATIVE PROBABILITIES

Time	Probability
11:02:11	69.8829%
11:02:51	84.8817%
11:03:31	84.9262%
11:05:31	97.2994%
11:06:11	99.9550%
11:06:51	99.9629%
11:08:51	99.9932%
11:09:31	99.9998%
11:10:11	99.9998%

The reason that cumulative probability never reaches 100% percentage is the *max_transmission_rounds* and *probability_threshold* filters that limit the number of considered rounds. If those limits were raised (i.e., more transmission rounds, smaller probability threshold), the result would consider cases of 4 or more transmission rounds, and the new percentages would be summed up to a percentage closer to 100%. The absolute 100% can be theoretically achieved if the *max_transmission_rounds* parameter is set to infinite and *probability_threshold* is set to zero.

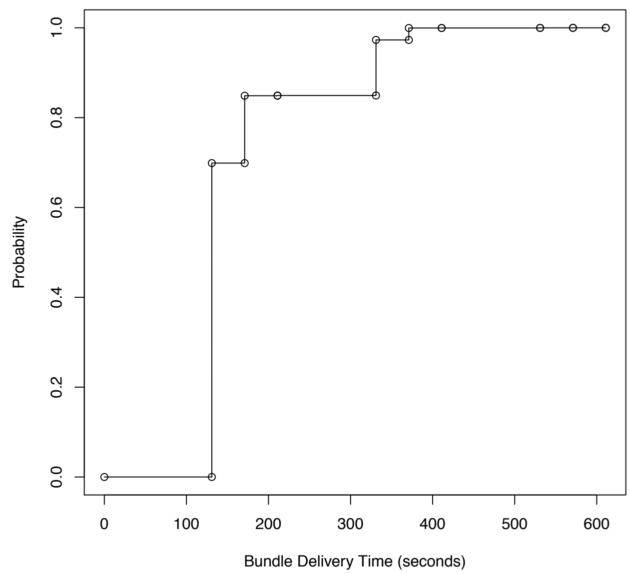


Fig. 3. Bundle delivery times and corresponding probabilities.

One of the useful results extracted from the application output is the earliest plausible arrival time. The latest plausible arrival time is theoretically infinite; in practice, however, it is the end of all contact possibilities between the communicating network nodes. Additionally, the possibility that the bundle will not reach its destination can also be calculated based on bundle's TTL, on the calculations, and on the contact plan.

Based on the results of the figure, we can also use the application as a QoS equivalent for space communications. Given a certain confidence *C* as user input, BDTE can estimate the time that its delivery is guaranteed with confidence *C*. For example in our scenario for a confidence input of 95%, we can guarantee that a bundle will have reached its destination with 95% confidence before 11:05:31. For this calculation we consider the cumulative probability that is greater or equal to *C*, since for the previous time (i.e., 11:03:31) we can't guarantee the delivery with confidence 95%.

In addition, if we use a specific time in the future as application input, we can find the probability that the bundle will have reached the destination before that time. For example, if user input is 11:04:00 in our scenario, BDTE can guarantee delivery of this bundle before 11:04:00 with a confidence of ~84.9%.

V. OPEN ISSUES

During our study we have identified a number of open issues, which we briefly present in the following paragraphs.

In case of a best-effort bundle transmission, queuing delay should not be always neglected, as it could be proportional or even bigger than the other latency components. Nevertheless, there is no straightforward solution to the queuing delay estimation problem, since it requires a different analysis and prediction of the network nodes' behavior in the future. However, this concept alone requires separate study.

During the database information description, we have assumed that network snapshots are taken periodically and thus form a time series with equal observation intervals. This is not always possible in reality, as space assets perform other critical tasks as well and thus, the network snapshots in the database might not represent equal time slots. BER statistics should therefore be modeled in a way to cope with realistic instrumentation, including unequal time series steps, intervals with no information at all or links with minimum available information.

Algorithm complexity can increase execution time, especially for plans with large numbers of contacts. It should be simplified and optimized in case it is applied in a distributed fashion. Such optimization is beyond the scope of this paper. Furthermore, since the BDTE application is intended to be executed in an administrative node for an off-line time delivery analysis and not for real-time decisions, then BDTE in its current form is appropriate indeed.

Finally, our method could use other routing algorithms as well, apart from CGR, with the proper adjustment for each scheme, since the functionality of each algorithm may vary.

VI. CONCLUSION

In this paper we have presented a novel method of estimating bundle delivery time in space telecommunications, using the CGR route computation algorithm. Our technique is based on an instrumentation database that stores statistics for each network node. Past BER values are calculated through some simple metrics extracted from the DB. Future BER values are then predicted via a Holt-Winters time series forecasting method and they are used in order to estimate the total number of transmission rounds that a bundle transmission will last. This procedure is executed successively for each hop. The result of our algorithm is a list of plausible bundle delivery times at destination with the corresponding probabilities. Our method can be used for administrative purposes and can provide time delivery expectations for critical bundles. Earliest plausible delivery times may also be computed, and delivery probabilities prior to a given time in the future may be provided. Therefore, BDTE could provide a useful administrative tool to predict the performance of different space applications and adjust their functionality and usage in real-time.

REFERENCES

[1] *SISG Operations Concept for SSI - final version*, Interagency Operations Advisory Group, Space Internetworking Strategy Group, January 2011

[2] V. Cerf *et al*, "Delay-Tolerant Networking Architecture", IETF RFC 4838, Informational, April 2007.

[3] K. Scott and S. Burleigh, "Bundle Protocol Specification", IETF RFC 5050, Experimental, November 2007.

[4] M. Ramadas, S. Burleigh and S. Farrell, "Licklider Transmission Protocol - Specification", IETF RFC 5326, Experimental, April 2007.

[5] S. Burleigh, "Contact Graph Routing", IETF Internet Draft, Experimental, July 2010.

[6] Jet Propulsion Laboratory, Interplanetary Overlay Network, https://ion.ocp.ohiou.edu/.

[7] D.C Lee and W. Baek, "Expected file-delivery time of deferred NAK ARQ in CCSDS file-delivery protocol", Communications, IEEE Transactions on, vol.52, no.8, pp. 1408- 1416, Aug. 2004.

[8] W. Baek and D.C. Lee, "Analysis of CGSDS file delivery protocol: immediate NAK mode", Aerospace and Electronic Systems, IEEE Transactions on, vol.41, no.2, pp. 503- 524, April 2005.

[9] S. I. U and J. L. Gao, "CFDP performance over weather-dependent Ka-band channel", SpaceOps Conference, Rome, Italy, June 19 - 23, 2006.

[10] J. L. Gao and J. S. SeGui, "Performance evaluation of the CCSDS file delivery protocol - latency and storage requirement", Aerospace Conference, 2005 IEEE, pp.1300-1312, 5-12 March 2005.

[11] T. De Cola and M. Marchese, "Performance Analysis of Data Transfer Protocols over Space Communications", Aerospace and Electronic Systems, IEEE Transactions on, vol.41, no.4, pp. 1200-1223, October 2005

[12] R. Wang, B. L. Shrestha, X. Wu, T. Wang, A. Ayyagari, E. Tade, S. Horan, J. Hou, "Unreliable CCSDS File Delivery Protocol (CFDP) over Cislunar Communication Links", Aerospace and Electronic Systems, IEEE Transactions on, vol.46, no.1, pp. 147- 169, January 2010

[13] *CCSDS File Delivery Protocol (CFDP), Recommendation for Space Data System Standards*, CCSDS 727.0-B-4, January 2007.

[14] R. Wang, S. Burleigh, P. Parikh, Che-Jen Lin, Bo Sun, "Licklider Transmission Protocol (LTP)-Based DTN for Cislunar Communications", Networking, IEEE/ACM Transactions on, vol. 19, no. 2, pp. 359-368, April 2011.

[15] R. Wang, B.I. Gutha, and P.V. Rapet, "Window-Based and Rate-Based Transmission Control Mechanisms over Space-Internet Links", Aerospace and Electronic Systems, IEEE Transactions on, vol. 44, no. 1, pp. 157-170, January 2008.

[16] N. Bezirgiannidis, and V. Tsaoussidis, "Packet Size and DTN Transport Service: Evaluation on a DTN Testbed", International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, pp.1198-1205, 18-20 Oct. 2010.

[17] G. Papastergiou, N. Bezirgiannidis and V. Tsaoussidis, "On the Performance of Erasure Coding over Space DTNs", Wired/Wireless Internet Communication (WWIC) 2012, pp. 269-281, June 6-8, 2012

[18] E. S. Gardner, Jr., "Exponential Smoothing: The State of the Art", Journal of Forecasting, Vol. 4, 1-28, 1985.

[19] G. E. P. Box and G. M. Jenkins, "Time series analysis: Forecasting and control", San Francisco: Holden-Day, 1970.

[20] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages", International Journal of Forecasting, 2004, vol. 20, no. 1, pp. 5-10.

[21] P. R. Winters, "Forecasting Sales by Exponentially Weighted Moving Averages", Management Science, 1960, vol. 6, no. 3, pp. 324-342.

[22] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2012, http://www.R-project.org/.

[23] P. Turchin, "Complex Population Dynamics: A Theoretical/Empirical Synthesis", Princeton Univ. Press, 2003

Nikolaos Bezirgiannidis received a Diploma in Electrical and Computer Engineering from Aristotle University, Greece, and a M.Sc. degree in the scientific area of Computer Networks in the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece, in 2010. He is currently a PhD candidate under the advisory of Prof. Vassilis Tsaoussidis in the same department. His areas of study include delay tolerant networking, interplanetary internetworks and routing in space environments.

Scott Burleigh has been developing software at JPL since 1986. He was a co-author of the specifications for the Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) and Asynchronous Message Service (AMS), and as a member of the Delay-Tolerant Networking (DTN) Research Group of the Internet Research Task Force he co-authored the specifications for the DTN Bundle Protocol (RFC 5050) and the DTN Licklider Transmission Protocol for delay-tolerant ARQ (RFC 5326). He has developed implementations of these protocols that are designed for deep space mission systems, aiming to enable deployment of a delay-tolerant Solar System Internet.

Vassilis Tsaoussidis (M'96-SM'03) received a B.Sc. in Applied Mathematics from Aristotle University, Greece; a Diploma in Statistics and Computer Science from the Hellenic Institute of Statistics; and a Ph.D. in Computer Networks from Humboldt University, Berlin, Germany (1995). He held faculty positions in Rutgers University, New Brunswick, SUNY Stony Brook and Northeastern University, Boston. In May 2003, he joined the Department of Electrical and Computer Engineering of Democritus University, Greece, where he was elected full Professor in 2007. His research interests lie in the area of transport/network protocols, i.e., their design aspects and performance evaluation.

He was/is editor for IEEE Transactions on Mobile Computing, the Journal of Computer Networks the Journal of Wireless Communications and Mobile Computing the Journal of Mobile Multimedia and the International Journal of Parallel, Emergent and Distributed Systems. He participates(d) in several Technical Program Committees in his area of expertise, such as INFOCOM, GLOBECOM, ICCN, ISCC, EWCN, WLN, and several others.