

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 720671, 11 pages
doi:10.1155/2011/720671

Research Article

Experimental Evaluation of TCP-Based DTN for Cislunar Communications in Presence of Long Link Disruption

Ruhai Wang,^{1,2} Xuan Wu,² Qinyu Zhang,³ Tarik Taleb,⁴ Zhensheng Zhang,⁵ and Jia Hou¹

¹ School of Electronics and Information Engineering, Soochow University, Suzhou, Jiangsu 215006, China

² Phillip M. Drayer Department of Electrical Engineering, Lamar University, Beaumont, TX 77710, USA

³ Department of Electronics and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China

⁴ NEC Europe Ltd., NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany

⁵ Network Systems, Argon ST, San Diego, CA 92121, USA

Correspondence should be addressed to Ruhai Wang, wang@ee.lamar.edu

Received 2 May 2010; Accepted 4 September 2010

Academic Editor: Sergio Palazzo

Copyright © 2011 Ruhai Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Delay/disruption tolerant networking (DTN) technology is considered a new solution to highly stressed communications in space environments. To date, little work has been done in evaluating the effectiveness and performance of the available DTN protocols when they are applied to an interplanetary Internet, especially in presence of a long link disruption. In this paper, we present an experimental investigation of the DTN architecture with a Bundle Protocol (BP) running over TCP-based convergence layer (TCPCL) protocol in a simulated cislunar communication environment characterized by a long link disruption. The intent of this work is to investigate the effectiveness of the TCPCL-based DTN protocol in coping with long link disruptions, through realistic file transfer experiments using a PC-based test-bed. The experiment results show that the DTN protocol is effective in handling a long link disruption experienced in data transmission accompanied by a cislunar link delay and a high BER. The performance of the DTN is most adversely affected by link disruption time in comparison to the effect of link delay and BER. For the transmissions with a very long link disruption of hours, the variations in goodput are nominal with respect to the change in cislunar link delay.

1. Introduction

Several issues of the space internetworking are inappropriately addressed by TCP [1], limiting its performance in space communications, especially in deep space [2–4]. One major issue of TCP is that it is designed to use window-based transmission control algorithms. With the sliding window flow-control mechanisms, TCP regulates the amount of data a source can send by adjusting the window size in response to the acknowledgement information from the destination; the timeliness of that information is key to the effectiveness of the technique. Another major issue is that TCP cannot distinguish between data losses caused by network congestion and link errors. Long and variable round trip times (RTTs), frequent link disruption, high transmission errors rates, and asymmetric link bandwidth over space communications channels challenge the design of TCP.

Many alternative transmission control protocols have been developed for space communications [4–16]. These protocols were developed for applications in various levels of communication environments, so their focuses differ. Most of them [4–9, 11–15] focus on Earth-orbit satellite networks while a few [10, 16] are designed for deep-space interplanetary links. For a comparative summary of these protocols, please refer to Table I in [17].

A new communication architecture called delay tolerant networking (DTN) [16, 18] has recently been developed to combat the long link delay and frequent link disruptions that generally characterize space communications. DTN adopts a store-and-forward custody transmission mechanism to deal with challenging communication environments [19]. A store-and-forward overlay network is formed by a Bundle Protocol (BP) [20] operating at the application layer of the Internet architecture to provide custody-based, message-oriented transmission. BP enables message transmission

and reception by invoking the services of an underlying convergence layer protocol (CLP) [20, 21] stack; the nature of this stack may differ in different segments of any end-to-end DTN path. Currently, the TCP-based CLP (i.e., TCPCL) [21], user datagram protocol- (UDP-) based [22] CLP, Saratoga CLP [23], and Licklider transmission protocol (LTP, also known as “long-haul transmission protocol”) [24, 25] CLP are supported under BP. The National Aeronautics and Space Administration (NASA) is interested in deploying DTN for space and hopes to fly with it on space missions soon.

The DTN techniques and concepts have also been considered for use in other environments that exhibit similar communication characteristics, especially those subject to link disruption and those with long link delay. Some examples of these environments are terrestrial wireless networks, sensor-based networks, tactical/military adhoc network, disaster rescue and response networks, and underwater acoustic networks.

Some work has been done in studying DTN protocols and techniques. These works mainly focus on application, mobility modeling and routing in terrestrial wireless mobile networks and sensor-based networks [26–33]. A few works [34, 35] have been done to support the use of DTN architecture and protocols for applications in interplanetary communications. A DTN congestion control mechanism proposed in [34] is based on an economic pricing model. With this mechanism, DTN routers can autonomously decide whether to accept a bundle based on its local available information and the risk of accepting the bundle. A hop-by-hop local flow control mechanism [35] is introduced to DTN architecture to control buffer overflow, link saturation, and unnecessary retransmissions in IPN.

Some experimental work has recently been done on DTN’s application in space communications. The Deep Impact Network Experiment (DINET) of DTN [36] was conducted by JPL in 2008 using the EPOXI spacecraft located about 15 million miles from Earth as the first DTN router in space. The DINET project validated the use of DTN protocols in deep-space Internet. NASA also began operating DTN on the International Space Station (ISS) in July 2009, in a deployment led by the University of Colorado, Boulder (CU-Boulder) [37]. Experiments are also done with a United Kingdom disaster monitoring constellation (UK-DMC) satellite in low-Earth orbit which demonstrated data transfer from space using DTN BP [38].

In close collaboration with NASA JPL, extensive work has been done at Lamar University, Texas, to study DTN for space communications [39]. In this paper, we present an experimental investigation of the DTN architecture with a BP running over TCPCL protocol in an emulated cislunar communication environment characterized by a long link disruption. The experiment in this work was conducted through realistic file transfers by running the DTN protocol stack over a PC-based network test-bed. The intent of this work is to investigate the effectiveness of the TCPCL-based DTN protocols in presence of long link disruptions ranging from 30 minutes to 8 hours.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss DTN for space and TCP

convergence layer; in Section 3, we describe the experimental setup and configurations; we present the experimental results in Section 4; and we draw conclusions in Section 5 and suggest future work in Section 6.

2. DTN for Space and TCPCL

2.1. Performance Problems of TCP in Space. As introduced, the Internet TCP is designed based on some critical assumptions. Those assumptions do not hold in some emerging communication environments, resulting in severe performance degradation. The impact on performance of TCP caused by these issues is serious in challenging space communication environment. Frequent and length link outages, long and variable propagation delays, strong channel noise in space all conspire to adversely degrade the performance of TCP.

Planetary bodies, asteroids, or spacecrafts moving in space all may periodically and randomly interrupt the communication link and cause frequent and long link outage between two communication endpoints [3]. This can cause a significantly large number of packet losses and retransmissions, resulting in severe performance degradation of TCP. Round-trip delays involved in space communications, especially those in deep space, are much longer than the terrestrial Internet. Long end-to-end RTTs in space communications hurt TCP interactions between the space node and the ground Internet node. As a result, this limits the usefulness of TCP acknowledgment information from the remote destination node and influences the effectiveness of TCP transmission control. See [40] for a detailed discussion of the impact of link disruption, delay, and other factors.

2.2. DTN and BP for Space. As the Internet protocols (mainly TCP/IP) are not effective across frequently disrupted links, long and variable delay, and high rates of data loss due to corruption, DTN technology is being considered for space communications. It is believed that integrating DTN into current space communication architecture will enable Internet-like user interaction to be maintained even in highly stressed space exploration environments.

DTN adopts a store-and-forward custody transmission mechanism to deal with challenging environments [18, 19]: each DTN node keeps a copy of every data packet sent until receiving an ACK confirming that the packet has been received successfully from the next node in the end-to-end path. This ensures that no data packets are lost even if a router is temporarily out of sight due to occultation or rotation in space.

As mentioned, BP is mainly designed to work with scheduled, predicted, and opportunistic connectivity for file transmission in presence of link interruption. It operates by invoking the services of an underlying CLP stack.

2.3. TCPCL. Currently, TCPCL, UDPCL and LTPCL are the most broadly supported CLPs under BP. The TCPCL for the DTN system works together with the well-known TCP to assure reliable communication services between DTN nodes

[21]. When a bundle node establishes a TCPCL connection, it establishes an underlying TCP connection to carry the TCPCL traffic. Over the established TCPCL connection, the sender can send bundles to the destination through the next node.

While TCP itself is a reliable protocol, the TCPCL adaptation is additionally capable of invoking various types of recovery mechanisms if the transfer of a bundle experiences interruption because of a TCP connection outage. For an idle connection, a one-byte keep-alive message may optionally be sent at a defined interval. The protocol uses this to detect loss of connection in the absence of bundle traffic. In [39], Wang et al. provide a side-by-side comparison between TCPCL and LTPCL to illustrate their design differences.

3. Experimental Setup and Configuration

A PC-based Space Communication and Networking Testbed (SCNT) [41] was built as an experimental platform at Lamar University to implement a typical cislunar and interplanetary communications infrastructure. The testbed was built based on the core design of our space link simulator (SLS) [42] and extended to perform relay communications. As illustrated in [41], the test-bed consists of three Linux-based PCs and the SLS—a source PC called TX (*tx.lamar.edu*), a relay PC called MX (*relay.lamar.edu*), and a destination PC called RX (*rx.lamar.edu*). A file transfer from the source PC, through the relay PC, to the destination PC emulates data transfer from the Moon to the Earth through a lunar orbiter. Previous research [17, 43–45] shows that the evaluation results obtained from the SCNT have generality, and the test-bed can fully evaluate the realistic performance of a protocol. Refer to [42] for a full description of SLS.

In our experiment, data is transferred via BP/TCPCL over TCP/IP over the simulated cislunar communication links. The BP and TCPCL protocol implementation used is from DTN-2 Reference Implementation V2.5.0 [46]. The operating system running on all three PCs, TX, MX, and RX, is Red-Hat Enterprise Linux 4 (kernel 2.6.9).

A long link disruption is inevitable in cislunar and deep-space communications because of planetary bodies and/or spacecraft rotation and limited relay capability. We experiment with the DTN protocol suites over a cislunar link with long link disruption, that is, 30 minutes, 1 hour, 2 hours, 4 hours, and 8 hours, considered as the common link outages in cislunar communications. The data rates for both the return and forward channels are 115, 200 bit/s. We chose to study three BERs, 0, 10^{-6} , and 10^{-5} , to simulate error-free transmission, a moderate error rate, and the maximum acceptable error rate on the channel. The BERs are generated following a typical Additive White Gaussian Noise (AWGN) model [42].

For the same reason mentioned in [41], we choose nine different link delays to study –1280 ms, 1500 ms, 2000 ms, 2500 ms, 3000 ms, 3500 ms, 4000 ms, 4500 ms, and 5000 ms to evaluate the comprehensive performance of DTN in cislunar communication.

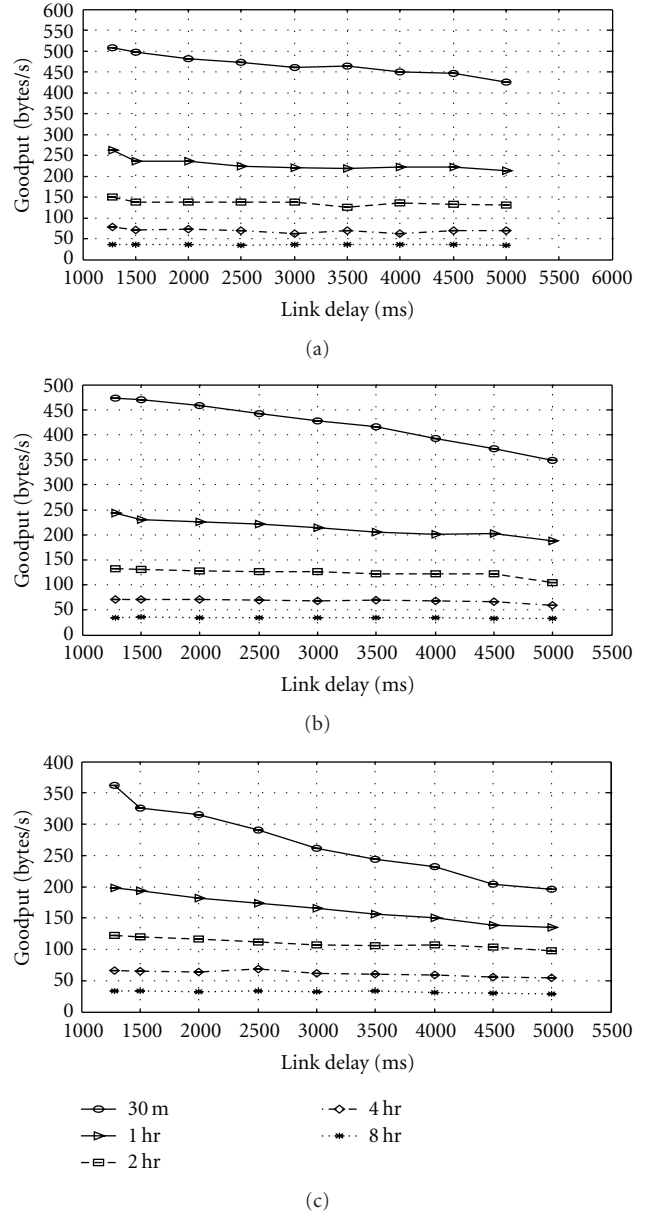


FIGURE 1: Goodput of DTN transmissions over cislunar channels with link disruption at different BERs. (a) BER = 0. (b) BER = 10^{-6} . (c) BER = 10^{-5} .

Similar to [41], the experiment was conducted by transferring data of 1,000,000 bytes (1 Mbyte) from the simulated source node on the Moon, TX, through the orbiter, MX, to the destination node on the Earth, RX, of the SCNT test-bed. Each file transfer was performed sixteen (16) times in each configuration. Network analysis toolkits including *Tcptrace* [47], *Xplot* [48], and *Jplot* [49] were used to analyze the performance of the protocol to support the evaluation.

4. Experiment Results and Discussion

In Figure 1, we compare the averaged goodput performance for the protocol in transmission of a 1 Mbyte file over

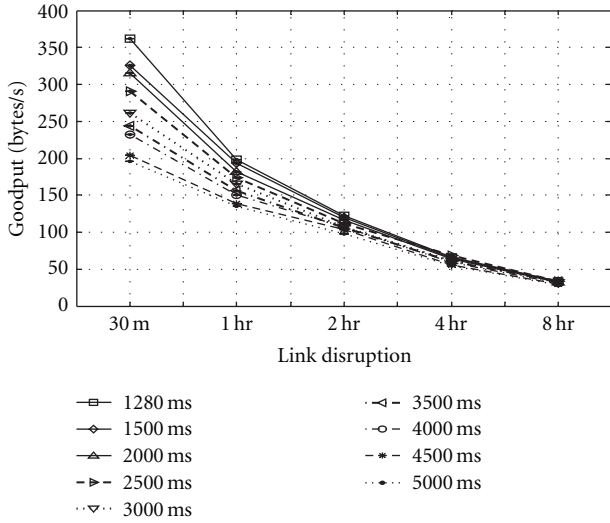


FIGURE 2: Comparison of goodput performance for DTN transmissions over cislunar channels for all link delays and link disruption at BER = 10^{-5} .

a simulated cislunar channels characterized by long link disruptions of 30 minutes ~ 8 hours. As we see, there is decrease in goodput with increase in link disruption time and BER. The goodput also has minor decreases with the increase in link delay time. However, given a BER, a significant decrease in goodput is observable for a moderately long link disruption (30 minutes ~ 1 hour) but not observable for a very long link disruption (2 hours ~ 8 hour). As link disruption time increases drastically, goodput decreases obviously. This is reasonable because with an increase in disruption time, DTN has to store more data in the relay orbiter, therefore delay increases, and consequently goodput drops. We also observe that for a long link disruption time, the variations in the good put are nominal with respect to the change of link delay. This happens because the link delays are so short compared with very long link break time that they do not have much effect on the goodput, making goodput remains the same for all delays. This is even more obvious if goodput versus link disruption is compared for each link delay, as illustrated in Figure 2 for the transmission at a BER of 10^{-5} . As different link delays make the protocols perform differently at a link break of 30 minutes, they cause negligible performance difference for a very long link disruption of 8 hours.

The impact of a long link disruption on the protocol performance is aggravated by a high BER, resulting in a large number of bundle retransmissions. This can be observed from a comparison of the number of retransmissions.

In Figure 3, the number of retransmissions is compared for the transmissions involving different link disruption time at different link delay and BER of 10^{-5} . As observed, the number of retransmissions is not only determined by a BER but also by the link disruption. With a given BER, the number of retransmissions increases exponentially along with the increase in link disruption. A hybrid of a high BER and link disruption results in a significantly

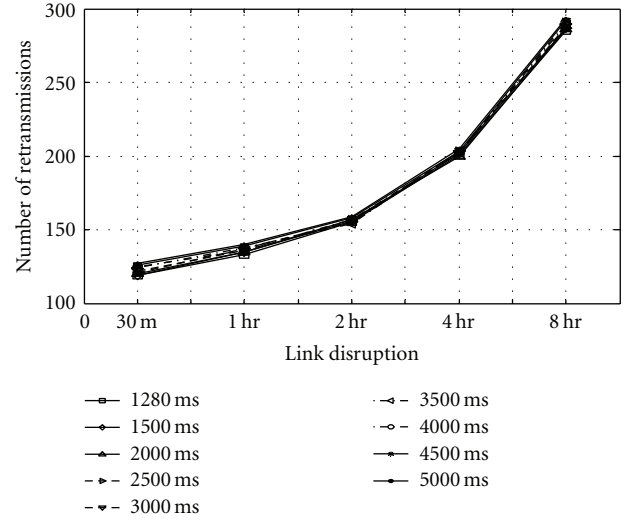


FIGURE 3: Comparison of number of retransmissions for DTN transmissions over cislunar channels with link disruption at a BER of 10^{-5} .

large number of retransmissions and thus decreases in the protocol performance. However, the change of link delay has a nominal effect on the number of retransmissions given a link disruption time and a BER. The transmissions at BERs of 0 and 10^{-6} have similar variation in the number of retransmissions and thus are not presented here.

Taking the transmission over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at BER = 10^{-6} , the operation of a typical DTN-transmission in the store-and-forward mode is investigated here. Figures 4 and 5 present a set of performance graphs, including time sequence graphs (TSGs) [47] and goodput graph, for the transmission from TX to the relaying MX.

Figure 4 depicts TSGs for the transmission of 1 Mbyte file from TX to the relaying MX. A TSG shows the transmission activity of packets, as well as events that happen during the lifetime of a connection by plotting the packet sequence numbers versus the file transfer time [47]. The segment-level TSG in Figure 4(a) shows a consistent, linear increase in the time sequence number, reflecting an ACK-based smooth transmission pattern of TCPCL-based DTN over a dedicated, point-to-point link. A vertical line segment with an arrow affixed to each end represents a packet sent by the sender; the stair-step at the bottom of the graph is the ACK line which tracks the acknowledgments returned by the receiver. As TX represents the source node on the Moon and MX represents the relay node in the cislunar orbit, there is only negligible link delay and channel BER between TX and MX. Provided that no link congestion is imposed on the channel, the packets are sent in a consistent and uniform pattern, until the transmission of 1 Mbyte file completes at around 18:41:26 as shown in Figure 4(b).

There is traffic of TCPCL keep-alive messages at MX prior to and after transmission of data packets. For an idle connection, the TCPCL keep-alive messages are sent periodically as one-byte packets to determine if the remote

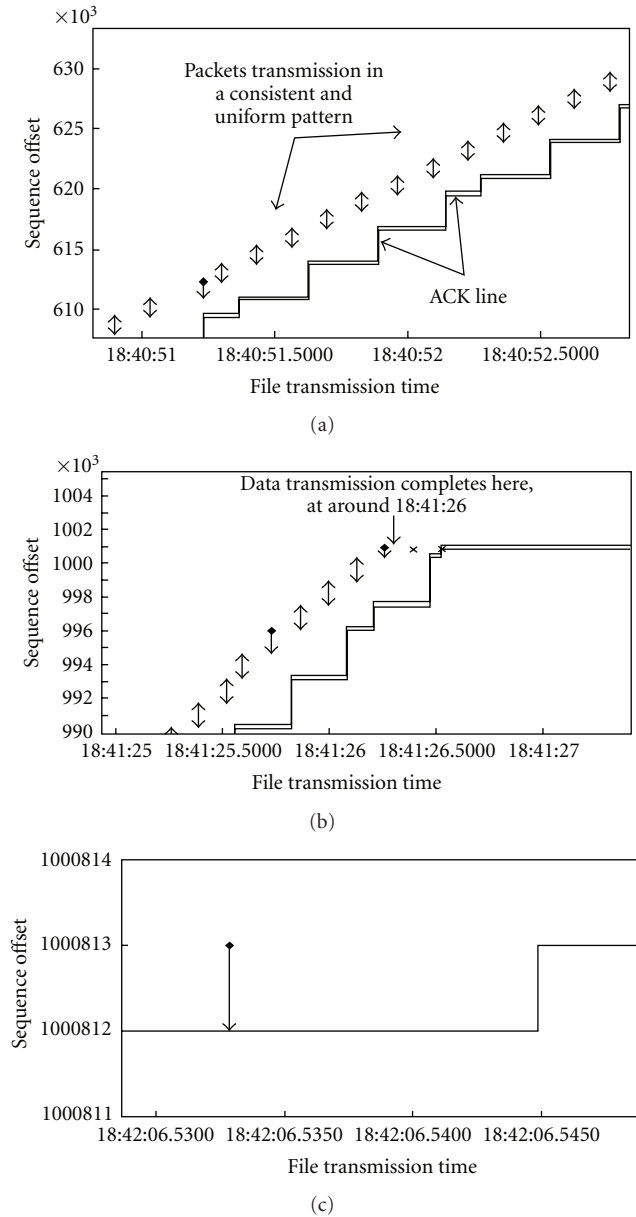


FIGURE 4: Time Sequence Graph (TSG) for DTN transmission of 1 Mbyte from TX to MX. (a) Consistent and uniform transmission pattern of DTN. (b) Completion of DTN data transmission. (c) One-byte TCPCL keep-alive message packet.

endpoint has either crashed or is down to prevent a long idle connection. When the TCPCL connection is established from TX to RX through relaying MX, a TCP connection from MX to TX is also built. After the TCPCL connection is established, for the time period before data transmission starts, the sending TCPCL at TX sends one-byte keep-alive message to keep track of the status of MX. The interleaving period of the messages is around 10 seconds. Similar keep-alive messages are transmitted after data transmission completes. When the data transmission starts, the source TX first transmits the message file to the relaying MX. Because of the design of store-and-forward operation, MX does not

start forwarding any message packets to the destination until the complete file is received from TX. This leaves the MX-RX connection idle during the file transmission from TX to MX, resulting in transmission of keep-alive message from MX to keep track of the liveness of the connection with RX. According to our observation, after RX receives the complete file, MX keeps sending keep-alive messages until it is manually stopped. This is believed to be specially designed feature of the DTN protocol. The keep-alive message packets are kept sending for possibly additional store-and-forward operation beyond RX. Figure 4(c) illustrates one (the fourth) of the keep-alive packet messages transmitted after data transmission. As seen, the one-byte keep-alive message is acknowledged by MX, which is a response to TX that the established connection is still alive.

Corresponding to the smooth increase in the TSG in Figure 4(a), an almost linear and consistent increase in goodput is observed in Figure 5 for the data transmission of 1 Mbyte from TX to MX. It turns out that the averaged goodput performance for the transmission is higher than 7100 bytes/s. After the complete file is transmitted, the goodput drops exponentially and approaches to 0 byte/s in response to the posttransmission period during which only one-byte keep-alive packets are transmitted by TX. The goodput remains at 0 byte/s until MX completes transmission of 1 Mbyte file to RX.

Figure 6 illustrate the TSGs for the transmission of 1 Mbyte file from MX to RX over a cislunar channel with link delay of 3000 ms, link disruption of 4 hours, and BER of 10⁻⁶. Figure 6 depicts the complete TSG trace for the transmission. Different from continuous and smooth pattern of the TSG in Figure 4(a), the transmission from MX to RX is divided into two separate segments by the very long link disruption of 4 hours in Figure 6. During the entire link disruption, the transmission of new packets is suspended and is not resumed until the disruption ends. A large number of SYN packets are transmitted during link disruption in an exponential and then uniform pace to probe the link connectivity.

During the first several minutes of link disruption, MX performs retransmission of a particular segment at an exponential pace. In fact, when the link is disrupted, the last packet sent prior to the outage was already on the way. Because of the link disruption, MX does not receive the acknowledgment for it. Therefore, MX resends this packet. After about five retransmissions at an exponential pace and without response, MX gives up. During the exponentially paced retransmissions, the first series of six one-byte SYN packets are also sent by MX to RX at an exponential pace to probe the link connectivity. During the link disruption of 4 hours, the sender cannot receive the ACK for the last data packet. After waiting for the ACK from RX for around 48 s, MX considers the link interrupted and sends SYN packets to request a new connection as a feature of TCP. Six SYN packets consist of one SYN packet followed by five retransmitted SYN packets represented as R SYN packets, with the time intervals between two packets also increasing exponentially, as illustrated in Figure 7. Once the link is resumed, MX starts transmitting the rest data until the whole file is sent out.

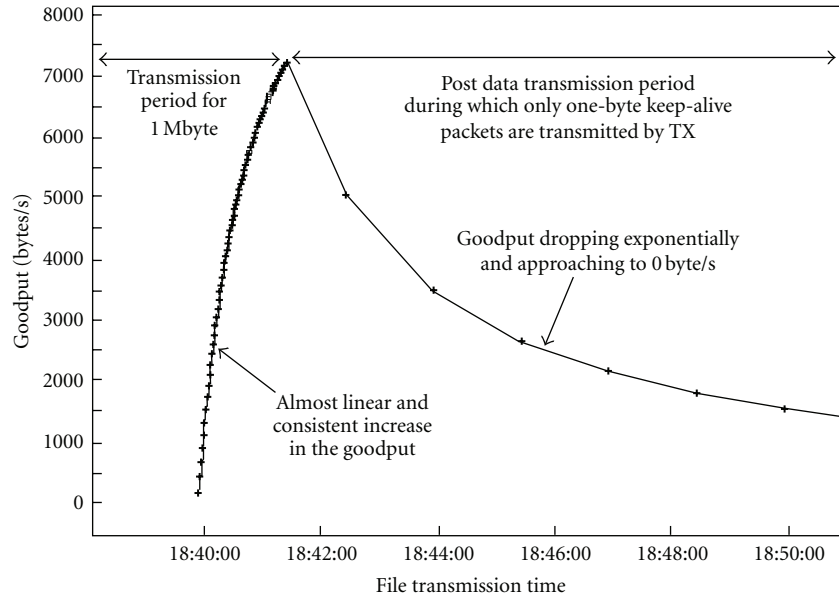


FIGURE 5: Goodput trace for DTN transmissions of 1 Mbyte from TX to MX.

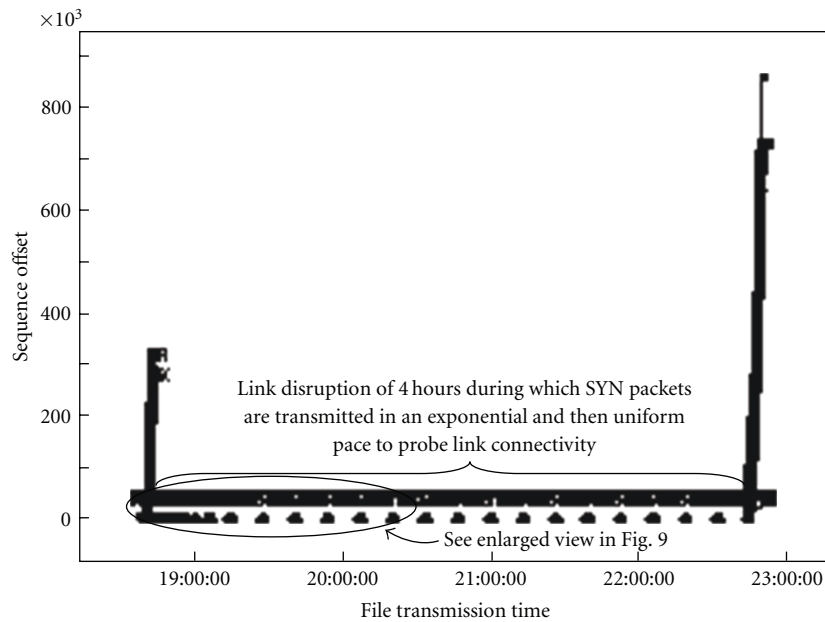


FIGURE 6: Time Sequence Graph for DTN transmission of 1 Mbyte file from MX to RX over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at BER = 10^{-6} .

Figure 8 illustrates an enlarged view of TSG startup for DTN data transmission from MX to RX. The first data packet is sent from MX to RX around 18:41:27 which is the time that TX just completes transmission of all the data packets to MX. This illustrates that whenever the file transmission over space channel is handled through a relay node, the relay node will not start forwarding any part of the file to destination until it receives the complete file from the source node or its previous relay node. In addition, the protocol starts the transmission from the initial window and increases the sending rate using the “Slow Start.” The graph reflects an exponential rise in

Congestion Window (CWND) in response to every packet acknowledged.

Figure 9 gives an enlarged view of the serial SYN packets that are sent during link outage from 18:44:00 to 20:20:00. As discussed in Figure 7, these series of SYN and R SYN packets are transmitted by MX to request a new connection. The series of SYN packets are sent at an exponential pace at the beginning. When the time interval between two series reaches to around 15 minutes, the pace changes to uniform. Each series consists of one SYN followed by five R SYNs, as in Figure 7.

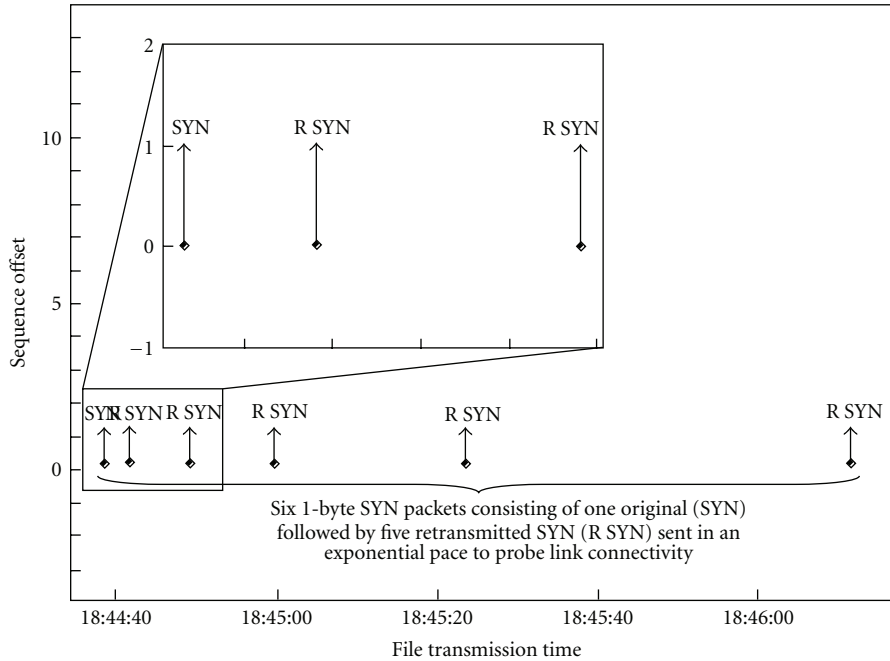


FIGURE 7: Time Sequence Graph for DTN transmission from MX to RX prior to link disruption over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at BER = 10^{-6} : enlarged view of first series of six 1-byte SYN packets sent by MX to RX during link disruption.

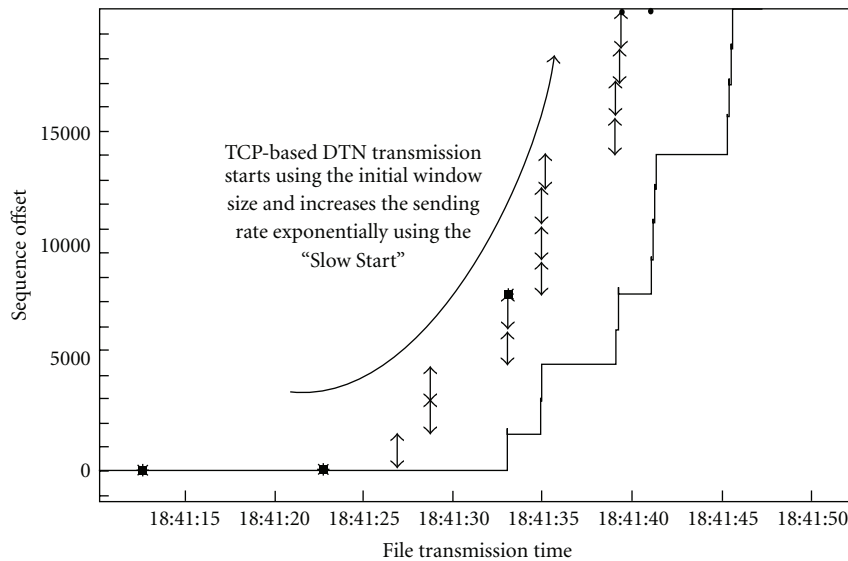


FIGURE 8: Enlarged view of TSG startup for DTN transmission from MX to RX over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at BER = 10^{-6} .

As soon as the disrupted link is resumed after 4-hour outage, MX continues transmitting the rest data until the whole file is sent out. As a general rule, after RX receives the complete file, MX keeps sending TCPCL keep-alive packets until it is manually stopped. This is similar to the transmission from TX to MX.

When bit errors are involved on the channel, packet corruptions and retransmissions happen in data transmission. With a BER of 10^{-6} involved, we find around eight

corruption/retransmission scenarios occurring during data transmission, which is correct according to the following calculation: $1\,000\,000 \times 8 \times 10^{-6} = 8$. Similarly, around 80 corruption scenarios are observed in DTN transmission with BER = 10^{-5} in our experiment, which is supported by the following calculation: $1\,000\,000 \times 8 \times 10^{-5} = 80$.

Figure 10 illustrates a typical traffic pattern of the TCPCL-based DTN transmission from MX to RX over cislunar channel with link delay of 3000 ms when there is

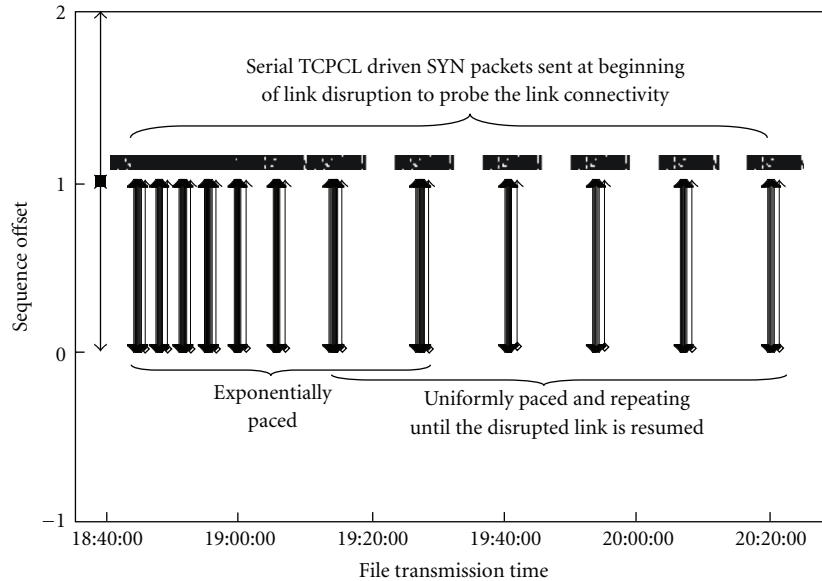


FIGURE 9: Enlarged view of serial SYN/R SYN packets sent from MX to RX during link disruption over cislunar channels.

no error scenarios occurring on channel. The pattern shows that the user file is transmitted in a series of packet flights, which is different from the transmission pattern between TX to MX in Figure 4. The packets in Figure 4 are sent one by one because of the negligible link delay. As observed, the transmission repeatedly pauses following the closure of the receiver window and then resumes as soon as the returning acknowledgment packets update both the edge of window and the ACK line, until all data is sent. This is a typical “send and wait for ACK” procedure with which the data transmission is mainly limited by the sender waiting for the long-delay ACK to open the receiver window.

Figure 10(b) depicts an enlarged view of the fifth flight in Figure 10(a). We see that a single flight contains ten packets. Each transmitted packet is not acknowledged (to the sender) until about 6.3 s after it is sent out. This time is dominated by a long RTT which is twice of the configured one-way link delay of 3000 ms. In other words, the sender has to wait at least entire RTT to receive the ACKs from the receiver. The ACK-clocked transmission mechanism involved in TCP-based DTN operation, thus, suffers severely from the long, idle time spent waiting for the ACK packets delivered from the receiver, as the standard TCP operates over any other long-delay channels. We also see that the data transmission has each ACK acknowledging two packets as it is configured with 1ACK/2 Segments by default.

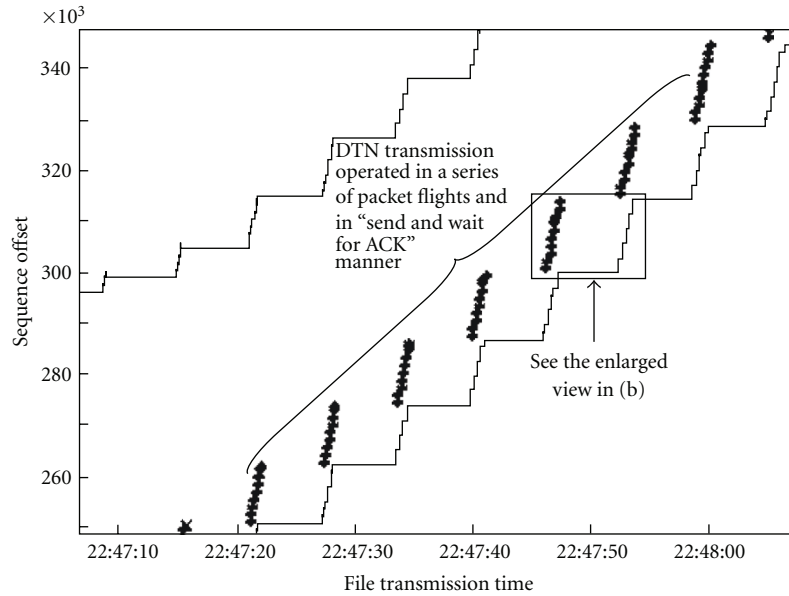
Corresponding to the TSGs in Figure 6, the goodput trace for the DTN transmission of 1 Mbyte from MX to RX over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at $\text{BER} = 10^{-6}$ is divided into two separate segments with no performance records between them. It has two separate segments because no data packets are transmitted during link disruption of 4 hours except for the SYN packets probing link connectivity. According to the experimental results, the averaged goodput for the

transmission is 67.96 bytes/s which is much lower than the throughput of 7195 bytes/s for the transmission between TX to MX, because of the impact of a very long link disruption.

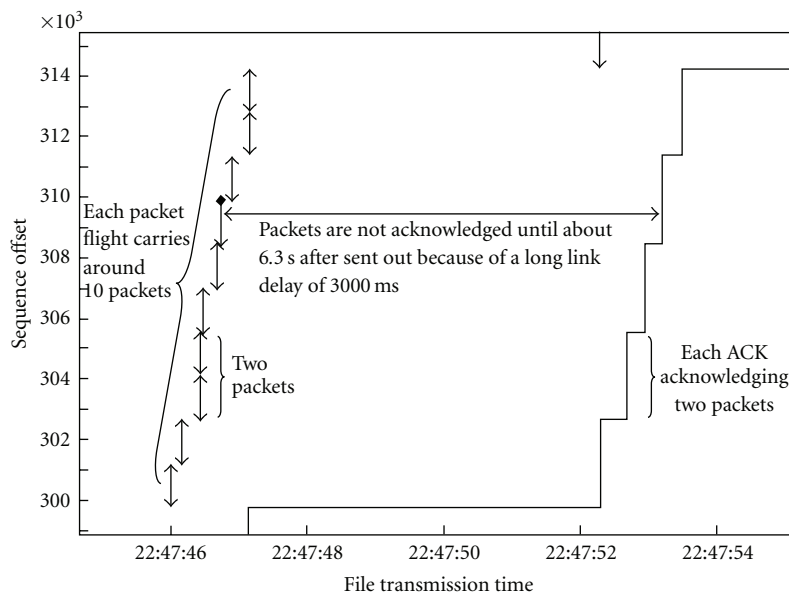
As another observation from the experiment results, the averaged goodput for the DTN transmission from MX to RX over cislunar channels with link delay of 3000 ms and link disruption of 4 hours at BER of 0 and 10^{-5} is 68.2 bytes/s and 61.17 bytes/s, compared with 67.96 bytes/s at BER of 10^{-6} . The goodput difference among the DTN transmissions at all three BERs (from 0 to 10^{-5}) is trivial because the effect of a high BER is so small that it does not have significant impact on the goodput compared to a very long link disruption time. This is different from the experimental results of the previous works [17, 44, 45] in which other space-Internet protocols (e.g., SCPS-TP and CFDP) are evaluated. In [17, 44, 45], because of no link disruption and/or with relatively short link outage involved during transmission, the goodput of both the rate-based and window-based protocols degrade significantly along with the increase of BER from 0 to 10^{-5} .

5. Conclusions

In this paper, we have presented an experimental evaluation of the DTN protocols with BP running over TCPCL over a long-delay cislunar communication channel characterized by a long link disruption. The major conclusions drawn are as follows. (1) The TCPCL-based DTN protocol is effective in handling a long link disruption (30 minutes ~ 8 hours) experienced in data transmission over a cislunar channel, even with an existence of a very long link delay of 1280 ms ~ 5000 ms and a high channel BER up to 10^{-5} . (2) The goodput performance of the DTN transmissions is most adversely affected by link disruption time in comparison to the effect of link delay and BER. There is roughly exponential decrease in



(a)



(b)

FIGURE 10: Traffic pattern of DTN transmission from MX to RX over cislunar channels with link delay of 3000 ms and link disruption of 4 hours without error. (a) DTN transmission in a series of packet flights. (b) Enlarged view of the fifth packet flight.

goodput along with exponential increase in link disruption time. (3) The impact of long link disruption on the protocol performance is aggravated by a high BER, resulting in a large number of bundle retransmissions. For a given BER, the number of retransmissions increases exponentially along with the increase in long link disruption. Given link disruption time and BER for a transmission, the change in link delay has a nominal effect on the number of retransmissions. (4) For transmissions with a particular BER, a significant decrease in goodput is observable for a moderately long link disruption (30 minutes ~ 1 hour) but not observable for a very long link disruption (2 hours ~ 8 hours). For a

long link disruption time, the variations in the goodput are nominal with respect to the change in link delay from 1280 ms to 5000 ms, as the link delays are so short, compared with very long link disruption time, that they do not have much effect on the goodput. (5) When the protocol stack runs over a long-delay cislunar channel without any link disruption, the ACK-clocked transmission mechanism in TCPCL-based DTN suffers severely from the long, idle time spent waiting for the ACKs and very slow error recovery with error-prone cislunar channels, similar to the operation of standard TCP over any other long-delay, lossy channel. (6) Different from the experimental results of the previous

work in which other space-Internet protocols (e.g., SCPS-TP and CFDP) are evaluated, the goodput difference among the DTN transmissions at all three BERs (from 0 to 10^{-5}) is trivial because the effect of a high BER is so small that it does not have significant impact on the goodput compared to a very long link disruption time. In comparison, the goodput of both the rate-based and window-based protocols in the previous work degrades significantly along with the increase in BER because of no link disruption and/or with relatively short link outage involved during transmission.

6. Future Work

As we concluded, the window-based transmission mechanism in the current TCPCL-based DTN-2 suffers severely from the long, idle time spent waiting for the ACKs and very slow error recovery with error-prone cislunar channels. According to our studies in the previous work [17], the traffic shaping mechanism of a rate-based transmission is much more effective than the bursty flow of window-based transmission over a long-delay, noisy space communication channel. However, current DTN-2 protocol implementation is not capable of controlling data transmissions in a rate-based manner. We plan to integrate a rate-based transmission mechanism into the current DTN-2 for its performance improvement over long-delay cislunar communications.

As mentioned, different CLPs (e.g., TCPCL, UDPCL, and LTPCL) are currently supported under BP. Each CLP is designed for a different type of communication link. Running any single CLP under BP may not be most efficient over a heterogeneous end-to-end interplanetary communication infrastructure. Some work has shown performance advantage of a hybrid of different transport layer protocols when no link disruption is involved [41]. Based on these preliminary results, we suggest that a hybrid of different DTN CLPs should be deeply investigated for efficient file transmission in interplanetary environment, that is, using one CLP for one hop of the end-to-end path and a different CLP for another hop of the path.

Acknowledgments

This paper was supported in part by the National Natural Science Foundation of China (NSFC) under Grant no. 61032003. The authors would like to thank the Associate Editors, Drs. Sergio Palazzo, Andrew T. Campbell, and Marcelo Dias de Amorim and the anonymous reviewers for their constructive comments and suggestions, which have helped to significantly improve the quality of this paper.

References

- [1] J. Postel, "Transmission control protocol—darpa Internet program—protocol specification," IETF Request for Comments RFC 793, September 1981.
- [2] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, 1997.
- [3] I. F. Akyildiz, Ö. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges," *Computer Networks*, vol. 43, no. 2, pp. 75–112, 2003.
- [4] R. C. Durst, G. J. Miller, and E. J. Travis, "TCP extensions for space communications," *Wireless Networks*, vol. 3, no. 5, pp. 389–403, 1997.
- [5] T. R. Henderson and R. H. Katz, "Transport protocols for internet-compatible satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 2, pp. 326–343, 1999.
- [6] M. E. Elaasar, M. Barbeau, E. Kranakis, and Z. Li, "Satellite transport protocol handling bit corruption, handoff and limited connectivity," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 2, pp. 489–502, 2005.
- [7] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: a new congestion control scheme for satellite IP networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 307–321, 2001.
- [8] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '02)*, pp. 89–102, August 2002.
- [9] K. Zhou, K. L. Yeung, and V. O. K. Li, "P-XCP: A-transport layer protocol for satellite IP networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 2707–2711, November 2004.
- [10] Ö. B. Akan, J. Fang, and I. F. Akyildiz, "TP-Planet: a reliable transport protocol for interplanetary internet," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 348–361, 2004.
- [11] T. Taleb, N. Kato, and Y. Nemoto, "An Explicit and Fair Window Adjustment Method to Enhance TCP Efficiency and Fairness Over Multihops Satellite Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 371–387, 2004.
- [12] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 467–479, 2002.
- [13] M. Marchese, M. Rossi, and G. Morabito, "PETRA: performance enhancing transport architecture for satellite communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 320–332, 2004.
- [14] M. Luglio, M. Y. Sanadidi, M. Gerla, and J. Stepanek, "On-Board Satellite "Split TCP" Proxy," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 362–370, 2004.
- [15] C. Roseti and E. Kristiansen, "TCP Noordwijk: TCP-based transport optimized for web traffic in satellite networks," in *Proceedings of the 26th International Communications Satellite Systems Conference (ICSSC '08)*, San Diego, Calif, USA, June 2008.
- [16] S. Burleigh, A. Hooke, L. Torgerson et al., "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.
- [17] R. Wang, B. Gutha, and P. V. Rapet, "Window-based and rate-based transmission control mechanisms over space-internet links," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 157–170, 2008.
- [18] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer*

- Communication (SIGCOMM '02)*, vol. 33, Karlsruhe, Germany, October 2003.
- [19] V. Cerf, S. Burleigh, A. Hooke et al., "Delay-tolerant networking architecture," IETF Request for Comments RFC 4838, April 2007, <http://www.ietf.org/rfc/rfc4838.txt>.
- [20] S. Burleigh and K. v., "Bundle protocol specification," IETF Request for Comments RFC 5050, November 2007, <http://www.ietf.org/rfc/rfc5050.txt>.
- [21] M. Demmer and J. Ott, "Delay tolerant networking TCP convergence layer protocol," IETF DTNRG IRTF Research Group, November 2008, <http://tools.ietf.org/html/draft-irtf-dtnrg-tcp-layer-02>.
- [22] J. Postel, "User datagram protocol," IETF Request for Comments RFC 768, August 1980.
- [23] L. Wood, W. M. Eddy, W. Ivanèic, J. McKim, and C. Jackson, "Saratoga: a delay-tolerant networking convergence layer with efficient link utilization," in *Proceedings of the International Workshop on Satellite and Space Communication (IWSSC '07)*, pp. 168–172, Salzburg, Austria, September 2007.
- [24] S. Burleigh, M. Ramadas, and S. Farrell, "Licklider transmission protocol—motivation," IETF Request for Comments RFC 5325, September 2008, <http://www.ietf.org/rfc/rfc5325.txt?number=5325>.
- [25] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider transmission protocol—specification," IETF Request for Comments RFC 5326, September 2008, <http://www.ietf.org/rfc/rfc5326.txt?number=5326>.
- [26] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, pp. 145–157, September 2004.
- [27] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM '05)*, pp. 1407–1418, Miami, Fla, USA, March 2005.
- [28] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '05)*, pp. 109–120, Philadelphia, Pa, USA, October 2005.
- [29] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: semantic models and routing algorithms," in *Proceedings of the ACM Workshops: Conference on Computer Communications (SIGCOMM '05)*, pp. 268–275, Philadelphia, Pa, USA, August 2005.
- [30] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proceedings of the ACM Workshops: Conference on Computer Communications (SIGCOMM '05)*, pp. 229–236, Philadelphia, Pa, USA, August 2005.
- [31] E. Oliver and H. Falaki, "Performance evaluation and analysis of delay tolerant networking," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiEval '07)*, pp. 1–6, San Juan, Puerto Rico, June 2007.
- [32] X. Zhang, J. K. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '07)*, pp. 195–206, Montreal, Canada, 2007.
- [33] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, 2007.
- [34] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous congestion control in delay-tolerant networks," in *Proceedings of 9th AIAA International Conference on Space Operations (SpaceOps '06)*, Rome, Italy, June 2006.
- [35] F. de Rango, M. Tropea, G. B. Laratta, and S. Marano, "Hop-by-hop local flow control over interplanetary networks based on DTN architecture," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 1920–1924, Beijing, China, June 2008.
- [36] V. Cerf, S. Burleigh, R. Jones, J. Wyatt, and A. Hooke, "First deep space node on the interplanetary internet: the deep impact networking experiment (DINET)," in *Presented at Ground System Architectures Workshop*, Torrance, Calif, USA, March 2009.
- [37] L. Clare, "Delay/disruption tolerant networking for space," in *Presented at Space-Enabled Global Communications & Electronic Systems Industry Update*, Cisco Systems, Irvine, Calif, USA, August 2009.
- [38] W. Ivancic, W. Eddy, L. Wood et al., "Delay/disruption-tolerant network testing using a LEO satellite," in *Proceedings of the 8th Annual NASA Earth Science Technology Conference (ESTC '08)*, University of Maryland, June 2008.
- [39] R. Wang, S. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider Transmission Protocol (LTP)-based DTN for cislunar communications," to appear in, *IEEE/ACM Transactions on Networking*.
- [40] R. Wang, X. Wu, T. Wang, and T. Taleb, "Delay tolerant networking (DTN) protocols for space communications," in *Delay Tolerant Networks: Protocols and Applications*, Auerbach Publications, CRC Press, September 2010.
- [41] R. Wang, X. Wu, T. Wang, X. Liu, and L. Zhou, "TCP convergence layer-based operation of DTN for long-delay cislunar communications," *IEEE Systems Journal*, vol. 4, no. 3, pp. 385–395, 2010.
- [42] S. Horan and R. Wang, "Design of a space channel simulator using virtual instrumentation software," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 5, pp. 912–917, 2002.
- [43] R. Wang and S. Horan, "Impact of Van Jacobson header compression on TCP/IP throughput performance over lossy space channels," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 2, pp. 681–692, 2005.
- [44] R. Wang, S. Horan, B. Tian, and S. Bonasu, "Optimal acknowledgment frequency over asymmetric space-internet links," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1311–1322, 2006.
- [45] R. Wang, B. Shrestha, X. Wu et al., "Unreliable CCSDS file delivery protocol (CFDP) over cislunar communication links," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 147–169, 2010.
- [46] DTN Reference Implementation, October 2007, <http://www.dtnrg.org/wiki/Code>.
- [47] TCP Connection Analysis Tool: TCPTRACE, <http://www.tcptrace.org/manual/index.html>.
- [48] T. Shepard, "TCP packet trace analysis," MIT-LCS-TR-494, MIT Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1991, <http://www.xplot.org/>.
- [49] Java Plotting Tool: jPlot, <http://masaka.cs.ohiou.edu/software/jPlot/>.