Performance of Bundle Protocol (BP) for deep space communications with long link disruptions

Ding Wang¹, Hongzhi Yu²

¹ Northwest Minzu University, Lanzhou, China
² Northwest Minzu University, Lanzhou, China

Corresponding author and e-mail: Ding Wang, wangding212@qq.com

Abstract. Near Earth missions and deep space missions will continue to be a major future space mission. A long link disruption is inevitable in space communications because of spacecraft rotation, planetary bodies and limited relay capability. A few works have been done with delay/disruption tolerant networking (DTN) technology for deep space communications and provided feasibility for its adoption in LEO- space and deep space missions. However, not much work has been done to fully evaluate the performance of DTN in such an environment, especially in the presence of custody and long link transfer. In this article, we present an experimental performance evaluation of DTN architecture and protocol stack with DTN custody transfer and long link break underneath bundle protocol (BP), in typical LEO-satellite and deep space communication infrastructures accompanied. The experiment was conducted by performing realistic file transfers over a PC-based test-bed.

1. Introduction

Bundle protocol is a protocol layer of DTN architecture [1]. It designed to work at the application layer of the DTN architecture. The protocol data unit communicated by this DTN bundle protocol is called Bundles and each bundle consist of two or more blocks of protocol data. [2] The node component that executes the programs of the BP and offers the BP services is known as the bundle protocol agent (BPA). The convergence layer adapter (CLA) utilizes the service of the underlying network protocol layers to send and receive the bundle on behalf of the BPA. [3]

Each DTN node keeps custody of bundles it forwards to the adjacent node until it receives a positive ACK from the adjacent node is so called store and forward mechanism. If the data is lost or if the node receives negative ACK, it re-forwards the bundle again. The custodian bundle is released when the node receives a notification that some other node or the particular destination node accepted the custody of the same bundle; or the bundle is deleted explicitly due to lifetime expiration. [3] So by this mechanism BP provided reliable transfer of data even in a highly stressed environment. Some of the other services provided by bundle protocol are error-free data delivery, completeness in delivery, fragmentation and extensibility [4].

Custody transfer is a good way to improve the reliability of network. Especially, it can help deliver a message to a moving responsibility destination. Protocol will request an acknowledgment to enhance the reliability. A message with custody transfer will not be deleted until it can be received by another node with custody transfer. Node will hold a message and is so called a custodian. Some custodians hold one message, some custodians own a message or message fragment. Protocol requests custody transfer to perform and deliver acknowledgment when host system can move the message to other nodes.
2. Experimental setup and configuration

This work focuses on performance evaluation of DTN in a LEO-satellite environment. LEO-satellite communication architectures have been well described in literature. For detail, refer to[5]. A PC-based Space Communication and Networking Testbed (SCNT) was built at Lamar University to emulate a relay-type of LEO-satellite and deep space communications infrastructure for the proposed experimental evaluation of DTN. Previous research[6] shows that the evaluation results obtained from the SCNT have generality and the test-bed can effectively evaluate the realistic performance of a protocol. A network emulation package of Linux OS, netem[7], was adopted to simulate the satellite channel conditions such as link outage, propagation delay, packet corruption, and packet loss. (http://www.linuxfoundation.org/collaborate/workgroups/networking/netem)

Table 1 illustrates a block diagram of SCNT. The test-bed consists of a Linux-based file source personal computer (PC) (tx.lamar.edu), file relay PC (relay.lamar.edu), file destination PC (rx.lamar.edu), and the SLS, which connects three file PCs.

| Protocol Layering | ION | BP/LTPCL/UDP |
|-------------------|-----|--------------|
| Custody transfer  | Have custody transfer | Have not custody transfer |
| Delay             | 300second |
| Link Breaks       | 10 min | 15 min | 30 min | 1 hour | 2 hour | 4 hour | 8 hour |
| Ratio of Channel Rate | 1:1  |
| BER               | 0 | 10^-6 | 10^-5 |
| File Size         | 10k | 100k | 1M | 10M |

3. Experimental results and discussions

In this section, we present and discuss the performance evaluation results of the LTP-based DTN protocol stack for deep space communication. A particular attention is on impact of BP custody transfer option.

3.1. BP/LTPCL with Custody Transfer vs BP/LTPCL without Custody Transfer with 1MB File Size

In Figure 1, a comparison of goodput performance for the BP/LTPCL with custody transfer vs BP/LTPCL without custody transfer to transmit a 1 Mbyte file over deep space channels is presented at three BERs with respect to no link outage and long link outages ranging from 10 minutes to 8 hours.

A direct observation is that for all the experimented transfer configurations (i.e., with custody transfer and without custody transfer) and bit error rate (0, 10^-5, and 10^-6), the transmissions without a link outage involved show significantly higher goodput than those with a link outage involved. While the goodput with a link break involved ranges from 3,000 bytes/s to 1,000 bytes/s, the goodput with a link break of 1 hour involved merges and approaches the same value. This is reasonable because a long link break dominates the short file transmission time, resulting in very poor goodput performance, provided that a high speed data channel rate was configured for transmission of a 1 M byte file during the experiment.
Figure 1. Goodput comparisons for the LTP-based DTN protocol to transmit a 1Mbyte files over deep space channels with different links disruption and BER. (a) BER=0 (b) BER=$10^{-6}$ (c) BER=$10^{-5}$

It is also observed that the performance is significantly different among two different transmission modes with custody transfer and without custody transfer for all the configurations when no link break is involved. In comparison, their performance merges and shows no significant differences for all link break durations varying from 2 hour to 8 hours.

As another observation, for all two transmission modes, the goodput performance without a link break decreases when either a packet bit error rate of $10^{-5}$ and/or $10^{-6}$ are introduced. This can also be easily explained. Data loss causes retransmission of data packets (although in a deferred mode), creating additional file transmission time thus degradation of goodput. When bit error rate is involved, the forward channel rate may be too low to handle the transmission of returning ACKs effectively, resulting in late arrival or frequent losses of ACKs and consequentially performance degradation of the protocol.

The key reason for performance degradation when custody transfer is added to bundle communication over LTP is that it doubles the number of bundles that must be sent: for each original data bundle that is sent – via LTP reliability – in the forward direction, a small custody acceptance signal bundle is sent – again via LTP reliability – in the reverse direction. The total number of bytes is not doubled, because custody signals are much smaller than most original data bundles, but transmission of a custody signal causes an additional report segment and report ACK segment to be transmitted, so the LTP overhead is doubled.

Custody transfer does hop-by-hop reliable transmission while TCP does end-to-end reliable transmission. A single custody transfer hop may span multiple bundle forwarding hops, because not all nodes are required to take custody; the next custodian might not be an immediate neighbor in the network. A node that has taken custody of a bundle will typically re-forward the bundle if no custody acceptance signal has been received (for that bundle) from any other node prior to expiration of a “countdown timer” for that bundle.

The custody transfer system does not do any integrity checking (such as a checksum) on received bundles, but convergence-layer protocols will typically discard any received protocol data unit that is determined to be corrupt. A node would typically never receive the bundle transmitted by another node, because the convergence layer would have discarded it. If the convergence layer adapter did pass the corrupt bundle up to BP at the node, the bundle protocol agent could check its integrity and detect the corruption. In that event it might theoretically send back a custody refusal signal, but there really isn’t any custody transfer signal reason code that signifies to retransmit; reasons for custody refusal tend to be shortages of storage or forwarding resources.

From the above discussions, custody transfer really isn’t designed to be a NACK-based retransmission system. In fact, the only event that causes retransmission by a bundle custodian is expiration of a countdown timer. In general, custody transfer isn’t designed to be a full-featured reliability system at all. The specification imposes very few solid requirements on the implementer.

3.2. BP/LTPCL with Custody Transfer vs BP/LTPCL without Custody Transfer with 10MB File Size

In Figure 2, a comparison of goodput performance for the BP/LTPCL with custody transfer vs BP/LTPCL without custody transfer to transmit a 10 Mbyte file over deep space channels is presented at three BERs with respect to no link outage and long link outages ranging from 10 minutes to 8 hours. It illustrates a comparison of goodput performance for the transmissions with custody and without custody at three different bit error rates of $0$, $10^{-5}$ and $10^{-6}$. 
We observe that for all three BERs, the transmissions without custody transfer show significantly higher goodput than those with custody transfer and the performance advantage is more significant when no link break is involved. The biggest performance advantage is observed at a packet corruption rate of 0, and it gradually decreases with the increase of link break. The advantage drops significantly when a $10^{-6}$ error rate is introduced and is further decreased when the bit error rate is further increased to $10^{-5}$.

As we see, the goodput difference between custody transfer and no custody transfer at an error rate of 0 decreases from 3000 bytes/s to 1000 bytes/s and then to 500 bytes/s when the link disruption is increased from 0 to 10 minutes and then to 480 minutes. In comparison, the goodput difference without custody transfer decreases more significantly, from 2500 bytes/s to 800 bytes/s and then to only 440 bytes/s.

Figure 2. Goodput comparisons for the LTP-based DTN protocol to transmit a 10Mbyte files over deep space channels with different links disruption and BER. (a) BER=0 (b) BER=$10^{-6}$ (c) BER=$10^{-5}$

The comparison results indicate that the impact of deep space channel with custody and without custody transfer on the goodput performance of LTP-based DTN decreases along with the increase in link disruption. The impact decreases more significantly when link disruption (around 10 minutes) is occurred over the channel in comparison to no disruption. In other words, the impact of link disruption (with a rate around 10 minutes ~120 minutes) on the goodput performance of LTP increases for the transfer mode without custody. The impact increases more significantly over a deep space channel.

Figure 2 illustrates a comparison of goodput performance for the transmissions with bit error rates of 0, $10^{-5}$, and $10^{-6}$ at different file sizes for both transmissions with custody transfer and without custody transfer. For both transfer configurations, the transmissions with a bit error rate of 0 show significantly higher goodput than those with a bit error rate of $10^{-5}$, and the performance advantage is more significant for without custody transfer. This tells us that the impact of the experimented bit error rate (around $10^{-5}$) on the goodput performance of LTP is more significant without custody transfer than that with custody transfer.

3.3. BP/LTPCL with Custody Transfer vs BP/LTPCL without Custody Transfer without Break with Respect to File Size

In Figure 3, we compare the averaged goodput performance for the LTP-based DTN protocol to transmit the 10Kbyte 100Kbyte 1Mbyte and 10Mbyte file over deep space channels experiencing without long link disruption.

In Figure 3(a), the goodput performance is compared among two different transfer modes with bit error rate of 0. An obvious observation is that for an exponential increase in the different file size, there is an almost exponential increase in goodput. In fact, each time when the file size becomes ten times than before, the goodput increases to roughly 2.5 times. The goodput performance varies in the range of 9000 bytes/s to 400 bytes/s.

We also observe that the goodput is actually about the same between custody transfer or without custody transfer when the file size is the same except at 10Mbyte, regardless of the changes of bit loss rate, and transfer configuration (i.e. custody transfer or without custody transfer). This happens because a big file size of 10-1000kbyte is so small that they do not lead the transmission into steady step. Therefore, the goodput of the LTP-based DTN transmissions is not most adversely affected by custody transfer and without custody transfer. In other words, for a transmission over deep space channels with
no link disruptions, there is not much difference between custody transfer and without custody transfer in goodput performance of transmission except for a large file size (≥10Mbyte).

Figure 3. Goodput comparison for the LTP-based DTN protocol to transmit a 10Kbyte 100Kbyte 1Mbyte and 10Mbyte file over deep space channels with respect to no link outage and long link outages involved. (a) BER=0 (b) BER=10⁻⁶ (c) BER=10⁻⁵

The three different bit error rates have little noticeable goodput differences for big file size, especially at 1Mbyte and 10Mbyte, but they have a small disparity when the file size is around 1Mbyte~10Mbyte. This indicates that the experimented bit error rate from 0 to 10⁻⁶ has little impact on the performance of the protocol when the link outage is zero but it has no impact at all when the file size is around 100Kbyte or smaller because a small file size has less effect on the goodput than the experimented bit error rates.

For the transmissions with BER involved (i.e. with BERs of 10⁻⁶ and 10⁻⁵), the comparisons show very similar results, as in Figure 3 (a), we see a roughly exponential decrease in goodput along with the exponential increase in link outage duration. We observe that the performance of two different transfer mode configurations is quite similar except at big file size. This indicates that the custody transfer configuration has no impact at all on the performance of the protocol for a small file.

4. Conclusion
For deep space communications, BP/LTPCL protocol is very effective in data transmission in presence of an extremely long link delay of 5 minutes of one ways delay. The comparison results are different depending on the transmission condition. BP/LTP without custody transfer has performance advantage over with custody transfer when no break or over short break is involved for transmission of a big file. The comparisons with custody transfer and without custody transfer shows very similar performance for a long link break channel and/or small file sizes. BP/LTPCL without custody transfer to transmit big size file with long break is very effective. According to this study, custody transfer really isn’t designed to be a NACK-based retransmission system, and it does not show performance advantage in deep space communication.

Acknowledgment
The author would like to acknowledge Scott C. Burleigh at Jet Propulsion Laboratory (JPL) of California Institute of Technology for his support in configuring ION and conducting the experiment and many significant discussions. This paper is supported by “the Fundamental Research Funds for the central universities of Northwest Minzu University” (Grant No.31920190093)

References
[1] Ding Wang. “Performance of Licklider Transmission Protocol (LTP) in LEO-Satellite Communications with Link Disruptions” IEEE ICCICC16 Proceedings
[2] Burleigh, S. and K. Scott. 2007. Bundle protocol specification. Delay tolerant networking group.
[3] Warthman, Forrest. 2003. Delay-Tolerant Networks (DTNs): A tutorial. http://www.dtng.org/docs/tutorials/warthman-1.1.pdf (accessed June 2009).
[4] www.ietf.org/rfc/rfc5050.txt (accessed September 2010).
[5] R. Wang, B. Gutha, and P. K. Rapet, “Window-based and rate-based transmission control mechanisms over space-Internet links,” IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 1. January 2008, pp. 157-170.
[6] R. Wang, S. Burleigh, P. Parik, C-J Lin, and B. Sun, “Licklider Transmission Protocol (LTP)-based DTN for cislunar communications,” IEEE/ACM Transactions on Networking, vol. 19, no.
2, April 2011, pp. 359-368. [Online]. Available: 
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5671516.

[7] Netem: network emulation package of Linux OS, [Online]. Available: 
http://www.linuxfoundation.org/collaborate/workgroups/networking/netem