Network Working Group   F. Templin, Ed.
Internet-Draft   Boeing Research & Technology
Intended status: Informational   April 10, 2020
Expires: October 12, 2020

LTP Fragmentation
draft-templin-dtn-ltpfrag-00

Abstract

The Licklider Transmission Protocol (LTP) provides a reliable datagram "covergence layer" for the Delay/Disruption Tolerant Networking (DTN) Bundle Protocol. In common practice, LTP is often configured over UDP/IP sockets and inherits its maximum segment size from the maximum-sized UDP datagram. This document discussses LTP interactions with IP fragmentation and mitigations for managing the amount of IP fragmentation employed.

Table of Contents

1. Introduction

The Licklider Transmission Protocol (LTP) [RFC5326] provides a reliable datagram "covergence layer" for the Delay/Disruption Tolerant Networking (DTN) Bundle Protocol (BP) [I-D.ietf-dtn-bpbis]. In common practice, LTP is often configured over UDP/IP sockets and inherits its maximum segment size from the maximum-sized UDP datagram (i.e. 2^16 bytes minus header sizes).

LTP breaks BP bundles into "blocks", then further breaks these blocks into "segments". The segment size is a configurable option and represents the largest atomic block of data that LTP will require underlying layers to deliver as a single unit. The segment size is therefore also known as the "retransmission unit", since each lost segment must be retransmitted in its entireity.

When LTP presents a segment to the operating system kernel (e.g., via a sendmsg() system call), the UDP layer frames the segment in a UDP header. The UDP layer then presents the resulting datagram to the IP layer for packet framing and transmission over a networked path. The path is further characterized by the path Maximum Transmission Unit (Path-MTU) which is a measure of the smallest link MTU (Link-MTU) among all links in the path.

When LTP presents a segment to the kernel that is larger than the Path-MTU, the IP layer performs IP fragmentation to break the datagram into fragments that are no larger than the Path-MTU. For example, if the LTP segment size is 64000 bytes and the Path-MTU is 1280 bytes IP fragmentation results in 50+ fragments that are transmitted as individual IP packets.

Each IP fragment is subject to the same best-effort delivery service offered by the network according to current congestion and/or link signal quality conditions; therefore, the IP fragment size becomes known as the "loss unit". Especially when the packet loss rate is considerable, however, performance can suffer dramatically when the loss unit is significantly smaller than the retransmission unit. In particular, if even a single IP fragment of a fragmented LTP segment is lost then the entire LTP segment is deemed lost and must be retransmitted.

This document discussses LTP interactions with IP fragmentation and mitigations for managing the amount of IP fragmentation employed.

2. Terminology

IETF keywords per [RFC2119] are not applicable within the scope of this document.

3. LTP Fragmentation

In common LTP implementations over UDP/IP (e.g., the Interplanetary Overlay Network (ION)), performance is greatly dependent on the LTP segment size. This is due to the fact that a larger segment presented to UDP/IP as a single uint incurs only a single system call and a single data copy from application to kernel space via the sendmsg() system call. Once inside the kernel, the segment incurs UDP/IP encapsulation and IP fragmentation which again results in a loss unit smaller than the retransmission unit. However, during fragmentation, each fragment is transmitted immediately following the previous without delay so that the fragments appear as a "burst" of consecutive packets over the network path resulting in high network utilization.

In order to avoid retransmission congestion (i.e., especially when the loss probability is non-negligible), the natural choice would be to set the LTP segment size to a size that is no larger than the Path-MTU. However, transmission of 64KB of data using a 1280B segment size would require 50+ independent sendmsg() system calls and data copies as opposed to just one when the largest segment size is used. This greatly reduces the bandwidth advantage offered by IP fragmentation bursts. Therefore, a means for providing the best aspects of both large segment fragment bursting and small segment retransmission efficiency is needed.

Fortunately, common operating systems such as linux provide a facility such as the sendmmsg() ("send multiple message") system call that allows the LTP application to present the kernel with a vector of segments instead of just a single segment. This affords the

bursting behavior of IP fragmentation coupled with the retransmission efficiency of employing small segment sizes.

This work therefore recommends implementations of LTP to employ a large block size, a conservative segment size and a new configuration option known as the "Burst-Limit" which determines the number of segments that can be presented in a single sendmmsg() system call. When the implementation receives an LTP block, it carves Burst-Limit-many segments from the block and presents the vector of segments to sendmmsg(). The kernel will prepare each segment as an independent UDP/IP packet and transmit them into the network as a burst in a fashion that parallels IP fragmentation. The loss unit and retransmission unit will be the same, therefore loss of a single segment does not result in a retransmission congestion event.

It should be noted that the Burst-Limit is bounded only by the LTP block size and not by the maximum UDP datagram size. Therefore, bursts can in practice convey much more data than a singe IP fragmentation event. It should also be noted that the segment size can still be made larger than the Path-MTU in low-loss environments without danger of triggering retransmission storms. This would result in combined UDP message and IP fragment bursting for high network utilization in more robust environments. Finally, Burst-Limit need not be a static value and can adaptively increase or decrease according to time varying network conditions.

4. Implementation Status

A prototype implementation has been developed, and early testing is underway.

5. IANA Considerations

This document introduces no IANA considerations.

6. Security Considerations

Communications networking security is necessary to preserve the confidentiality, integrity and availability.

7. Acknowledgements

The NASA Space Communications and Networks (SCaN) directorate coordinates DTN activities for the International Space Station (ISS) and other space exploration initiatives.

Keith Philpott, Bill Pohlchuck and Eric Yeh are acknowledged for their significant contributions.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5326] Ramadas, M., Burleigh, S., and S. Farrell, "Licklider Transmission Protocol - Specification", RFC 5326, DOI 10.17487/RFC5326, September 2008, <https://www.rfc-editor.org/info/rfc5326>.

8.2. Informative References

[I-D.ietf-dtn-bpbis] Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol Version 7", draft-ietf-dtn-bpbis-24 (work in progress), March 2020.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org