July 17, 2023

A Guide to Configuring LTP for ION

Scott Burleigh, Jay Gao, Leigh Torgerson
Jet Propulsion Laboratory, California Institute of Technology

Version 4.1.3[1]

1 Introduction

ION's implementation of LTP is challenging to configure: there are a lot of configuration parameters to set, because the design is intended to support a very wide variety of deployment scenarios that are optimized for a variety of different figures of merit (utility metrics).

LTP-ION is managed as a collection of "spans", that is, transmission/reception relationships between the local LTP engine (the engine – or DTN "node" – that you are configuring) and each other LTP engine with which the local engine can exchange LTP protocol segments. Spans are managed using functions defined in libltpP.c that are offered to the operator by the ltpadmin program.

ltpadmin can be used to add a span, update an existing span, delete a span, provide current information on a specified span, or list all spans. The span configuration parameters that must be set when you add or update a span are as follows:

- The **remote LTP engine number** identifying the span. For ION, this is by convention the same as the BP node number as established when the ION database was initialized.
- The **maximum number of export sessions** that can be held open on this span at any one time. This implements LTP flow control across the span: since no new data can be transmitted until it is appended to a block – the data to be conveyed in a single export session – and no new session can be started until the total number of open sessions drops below the maximum, the closure of export sessions regulates the rate at which LTP can be used to transmit data.
- The **maximum number of import sessions** that will be open on this span at any one time. This value is simply the remote engine's own value for the "maximum number of export sessions" parameter.
- **Maximum LTP segment size**. This value is typically the maximum permitted size of the payload of each link-layer protocol data unit – nominally a *frame*.
- **Aggregation size limit**. This is the "nominal" size for blocks to be sent over this span: normally LTP will concatenate multiple service data units (such as BP bundles) into a single block until the aggregate size

[1] New version number now tracks with the ION software release cycle. Therefore, the previous version number (v6) is now obsolete and replaced by the ION version number for which it will be released on SourceForge.

of those service data units exceeds the aggregation size limit, and only then will it divide the block into segments and use the underlying link service to transmit the segments. (Note that it is normal for the aggregation size limit to be exceeded. In this sense, the word "limit" is really a misnomer; "threshold" would be a better term.)

- **Aggregation time limit**. This parameter establishes an alternate means of terminating block aggregation and initiating segment transmission: in the event that service data units are not being presented to LTP rapidly enough to promptly fill blocks of nominal size, LTP will arbitrarily terminate aggregation when the length of time that the oldest service data units in the block have been waiting for transmission exceeds the aggregation time limit.
- **The Link Service Output command.** This parameter declares the command that will be used to start the link service output task for this span. The value of this parameter is a string, typically enclosed in single quote marks and typically beginning with the name of the executable object for the task. When the "udplso" link service output module is to be used for a given span, the module name is followed by the IPAddress:Port of the remote engine and (optionally) the UDP transmission rate limit in bits per second.

In addition, at the time you initialize LTP (normally at the start of the ltpadmin configuration file) you must set one further configuration parameter:

- **Estimated total number of export sessions**, for all spans: this value is used to size the hash table that LTP uses for storing and retrieving export session information.

In many cases, the best values for these configuration parameters will not be obvious to the DTN network administrator. To simplify this task, an LTP Configuration Worksheet has been developed.

2 Worksheet overview

The LTP configuration worksheet is designed to aid in the configuration of a single span <u>at both ends</u> – that is, the worksheet for the span between engines X and Y will provide configuration parameter values for use in commanding ltpadmin on both engine X and engine Y.

The cells of the worksheet are of two general types, Input Cells and Calculated Cells:

- Input Cells are cells in which the network administrator must supply values based on project decisions. These cells are yellow-filled.
- Calculated Cells are cells that are computed by the worksheet based on LTP configuration principles. These cells are grey-filled. The cells are protected from modification (though you can unprotect them if you want by selecting "Unprotect Sheet" on the Excel "Review" tab).

Some of these cells are used as span configuration parameters or are figures of merit for network administrators:

- Span configuration parameters are identified by an adjacent dark grey title cell with white text.
- Figures of merit for which the network administrator may want to optimize the span configuration are identified by an adjacent green title cell with red italic text.

Note: *Configuration parameters that are described in detail in this document are numbered. To ease cross referencing between this document and the worksheet, the parameter numbers are placed next to the title cells in the worksheet.*

3 Input Parameters

This section provides guidance on the values that must be supplied by the network administrator. Global parameters affect calculated values and configuration file parameters for all spans involving the local LTP engine.

3.1 Global parameters

Maximum bit error rate is the maximum bit error rate that the LTP should provide for in computing the maximum number of transmission efforts to initiate in the course of transmitting a given block. (Note that this computation is also sensitive to data segment size and to the size of the block that is to be transmitted.) The default value is .000001, i.e., 10^{-6}, one uncorrected (but detected) bit error per million bits transmitted.

The **size (mean) of an LTP report segment (in bytes)** may vary slightly depending on the sizes of the session numbers in use. 25 bytes is a reasonable estimate.

3.2 Basic input parameters

Values for the following parameters must be provided by the network administrator in order for the worksheet to guide the configuration. Values must be provided for both engine “X” and engine “Y”.

0. The OWLT between engines (sec) is the maximum one-way light time over this span, i.e., the distance between the engines. (Note that this value is assumed to be symmetrical.)
1. A unique engine number for each engine.
2. The IP address of each engine. (Assuming udplso will be used as the link service output daemon.)
3. The LTP reception port number for each engine. (Again assuming udplso will be used as the link service output daemon.)
4. An estimate of the mean size of the LTP service data units (nominally bundles) sent from this engine over this span.
5. Link service overhead. The expected number of bytes of link service protocol header information per LTP segment.
6. Aggregation size limit (described above). Note that a suggested value for this parameter is automatically computed as described below, based on available return channel capacity.
7. The scheduled transmission rate (in bytes per second) at which this engine will transmit data over this span when the two engines are in contact.
8. Maximum percentage of channel capacity that may be consumed by LTP report segments. A warning will be displayed if other configuration parameters cause this limit to be breached.
9. An estimate of the percentage of all data sent over this span that will be “red” data, i.e., will be subject to positive and negative LTP acknowledgment.
10. Aggregation time limit. The minimum value is 1 second. Increasing this limit can marginally reduce the number of blocks transmitted, and hence protocol overhead, at times of low communication activity. However, it reduces the “responsiveness” of the protocol, increasing the maximum possible delay before transmission of any given service data unit. (This delay is referred to as “data aggregation latency”.)
    - “Low communication activity” is defined as a rate of presentation of service data to LTP that is less than the aggregation size limit divided by the aggregation time limit.
11. LTP segment size (bytes) is the maximum LTP segment size sent over this span by this engine. Typically, this is the maximum permitted size of the payloads of link-layer protocol data units (frames).
12. The maximum number of export sessions. This implements a form of flow control by placing a limit on the number of concurrent LTP sessions used to transmit blocks. Smaller numbers will result in slower

transmission, while higher numbers increase storage resource occupancy. Note that a suggested value for this parameter is automatically computed as described below, based on transmission rate and one-way light time.

4 Further Guidance

This section provides further information on the methods used to compute the Calculated Cells and also guidance for Input Cell values.

4.1 First-order computed parameters

The following parameters are automatically computed based on the values of the basic input parameters.

13. **Estimated “red” data transmission rate (bytes/sec)** is simply the scheduled transmission rate multiplied by the estimated “red” data percentage.
14. **Maximum export data in transit (bytes)** is the product of the estimated red data transmission rate and the round-trip light time (which is twice the one-way light time between the engines). This is the maximum amount of red data that cannot yet have been positively acknowledged by the remote engine and therefore must be retained in storage for possible retransmission.

4.2 Configuration decision parameters

Values for the following parameters must be chosen by the network administrator on the basis of (a) known project requirements or preferences. (b) the first-order computed parameters, and (c) the computed values of figures of merit that result from tentative parameter value selections, as noted.

6. **Aggregation size limit (revisited)**. Reducing this parameter tends to increase the number of blocks transmitted, increasing total protocol overhead. The suggested value for this parameter is computed as follows:
   - The maximum number of bytes of LTP report content that the remote engine may transmit per second is given by the product of the remote engine’s transmission data rate and the maximum percentage of the remote engine’s channel capacity that may be allocated to LTP reports.
   - The maximum number of reports per second transmitted by the remote engine is the maximum number of LTP report content bytes transmitted per second divided by the mean report segment size.
   - Assuming that normally all blocks are received without error, the maximum number of blocks to be transmitted per second by the local engine should be equal to the maximum number of reports that may be transmitted per second by the remote engine.
   - The threshold block size, expressed in bytes per block, is then given by dividing the local engine’s transmission data rate (in bytes per second) by the maximum number of blocks to be transmitted per second.
15. **Est. mean export block size** is computed as follows:
    a. If the mean service data unit size is so large that aggregation of multiple service data units into a block is never necessary, then that mean service data unit size will in effect determine the mean export block size (one service data unit per block).
    b. Otherwise, the mean export block size will be determined by aggregation. If the red data transmission rate is so high that the aggregation time limit will normally never be reached, then the aggregation size limit constitutes the mean export block size. Otherwise, block size will be constrained by aggregation time limit expiration: the estimated mean export block size

will be approximated by multiplying the red data transmission rate by the number of seconds in the aggregation time limit.

   c. So estimated mean export block size is computed as larger of mean service data unit size and "expected aggregate block size", where expected aggregate block size is the lesser of block aggregation size limit and the product of red data transmission rate and aggregation time limit.

16. **Est. blocks transmitted per second** is computed as **Est. red data xmit rate (bytes/sec)** (parameter 13) divided by **Est. mean export block size** (parameter 15).
17. The **Est. report bytes/sec sent** by the remote engine in response to these transmitted blocks is computed as the product of **Est. blocks transmitted per second** (parameter 16) and **Size (mean) of LTP acknowledgment (bytes)** (a global parameter). When mean service data unit size is less than the aggregation size limit and the red data transmission rate is high enough to prevent the aggregation time limit from ever being reached, this value will be about the same as the maximum number of bytes of LTP report content that the remote engine may transmit per second as computed above. ***Note*** that increasing the aggregation size limit reduces the block transmission rate at the local engine, reducing the rate of transmission of acknowledgment data at the remote engine; this can be a significant consideration on highly asymmetrical links.
18. **Est. segments per block** is computed as **Est. mean export block size** (parameter 15) divided by **LTP segment size (bytes)** (parameter 11).
19. **Est. LTP delivery efficiency** on the span is calculated by dividing **Est. blocks delivered per second** by **Est blocks transmitted per second**. ***Note*** that reducing the aggregation size limit indirectly improves delivery efficiency by reducing block size, thus reducing the percentage of transmitted blocks that will be affected by the loss of a given number of frames.
12. **Maximum number of export sessions (revisited)**. Increasing the maximum number of export sessions will tend to improve link bandwidth utilization but will increase the amount of storage space needed for span state retention. The suggested value for this parameter is computed as the **maximum export data in transit (bytes)** (Parameter 14) divided by **Est. mean export block size** (parameter 15) as determined above. Configuring the span for a maximum export session count that is less than this limit will make it impossible to fully utilize the link even if all blocks are of estimated mean size.
20. **Nominal export SDU's in transit** is computed by dividing **Nominal export data in transit (bytes)** by the **Size (mean) of service data units (bytes)** (parameter 4).
21. **Expected link utilization** is then computed by dividing **Nominal export data in transit (bytes)** by **Maximum export data in transit (bytes)** (parameter 14). ***Note*** that a low value of expected link utilization indicates that a high percentage of the span's transmission capacity is not being used. Utilization can be improved by increasing estimated mean export block size (e.g., by increasing aggregation size limit) or by increasing the maximum number of export sessions.
22. **Max data aggregation latency (sec)** is simply the value supplied for **Aggregation time limit (sec)** (parameter 10) as this time limit is never exceeded.

4.3 LTP initialization parameters

Finally, the remaining LTP initialization parameter can be computed when all span configuration decisions have been made.

23. **Maximum number of import sessions** is automatically taken from the remote engine's maximum number of export sessions.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

5 Updated Features - May 2021

This section describes the following features added to the configuration tool as of May 2021:

1. A "link" worksheet has been added to set space link parameters such as frame size and error rate and to compute parameters such as *maxBer* and laboratory Ethernet-based frame loss simulation.

2. Conditional formatting has been added to a few entries in the *main* worksheet to provide visual cues for our-of-range parameters and warning messages to guide parameter selection.

3. A simple model to estimate the minimum required heapWord size for a one-hop LTP link.

5.1 Link worksheet

The recommended workflow for using the LTP configuration tool is to first establish the space link configuration using the *link* worksheet before attempting to generate a LTP configuration under the *main* worksheet. The *link* worksheet has the following input and computed cells:

- **Select CCSDS Frame Size (bits) [user input]–** this cell allows the user to select a standard CCSDS AOS/TM frame size from a drop down list that includes LDPC, Turbo, and Reed-Solomon codes.

- **CCSDS Frame Size (bytes) [computed]**– converts frame size from bits to bytes.

- **Desired Frame Error Rate [user input]** – this parameter sets the expected frame error rate of the LTP link in operation. This parameter could be derived from link budget analysis or a mission requirement document.

- **Segment size (byte) [user input]** – this parameter sets the maximum segment payload size used by LTP. The size of the segment, in relation to the underlying CCSDS frame, will determine the segment error rate and the probability that LTP will need to request retransmission.

- **Ethernet Frame Size (byte) [user input]** – this is the Ethernet frame size used in a laboratory environment to simulate space link frame losses.

- **Segment Error Rate Computation [computed]** – this is the LTP segment error rate derived from the frame error rate and the segment and CCSDS frame size selections.

- ***maxBER* Computation [computed]** – this is the computed *maxBER* parameter for LTP. The *maxBER* parameter is what LTP uses to estimate segment error rate, which in turn will affect how LTP handles handshaking failure and repeated retransmission requests. To properly operate LTP, the maxBER value provided must result in the same segment error rate as one expects to encounter in real space link.

- **Ethernet Error Rate Computation [computed]** – this is the recommended setting for using laboratory Ethernet frame error software/hardware to simulate space link loss. This value is translated from the segment error rate to Ethernet frame error to make sure that laboratory testing provides a statistically equivalent impact on LTP.

5.2 Updates to existing LTP calculation

In the *main* worksheet described in Section 3, we made the following enhancements:

- Item 6: **Aggregation size limit (bytes)** – a green icon is displayed when the input parameter is greater or equal to the suggested value; a red icon is displayed when this parameter is below the suggested value. The suggest value aggregation size limit upper bounds the LTP block rate such that the acknowledgement traffic (report segments) from the receiver to the sender can be supported.
- Item 10: **Aggregation time limit (sec)** – there are two factors affecting LTP block aggregation: time limit and size limit. The aggregation process stops as soon as one of the two limit is reached. A green icon is displayed if the time limit value in this cell is sufficiently large such that aggregation process will be size limited, i.e., on average the LTP block aggregation process will reach the size limit before the time limit. This is the nominal and desired configuration unless there is a strict latency requirement that forces one to use very low aggregation time limit. A red icon is displayed if the time limit will be driving, which means the LTP block size will generally be smaller than the aggregation size limit and the block rate will be higher than desired. If a latency requirement forces the use of low aggregation time limit, one must check to make sure there are still sufficient bandwidth to support the acknowledgement (report segment) traffic.
- Item 17: **Est. report bytes/sec sent** - this field estimates the bandwidth required to support LTP report segment traffic up to 95 percentile of all cases involving retransmission of missing segments. The segment error rate was derived from the *link* worksheet. The green icon indicates that estimated report bandwidth is feasible based on current configuration.

5.3 HeapWord Size estimate

A simple HeapWord size estimate calculation is added to the *main* worksheet, based on the following assumptions:

1. The only traffic flows in the system are those between node X and Y using LTP.
2. Heap is sized to support at least 1 contact session
3. Each contact starts with a clean slate. At the beginning of a contact, the heap space is not occupied by bundles/blocks/segments left over from previous contact or other unfinished processes.
4. Source user data is file-resident. Most ION test utility programs such as *bpsendfile* and *bpdriver* will keep source data (or create simulated data) in a file for as long as possible until just prior to transmission by the underlying convergence layer adaptor when pieces of user data are copied into each out-going convergence layer's PDUs. Please check how your software uses the BP API to determine how source data is handled. If in doubt, you may need to increase the heap space allocation to hold the user's source data.
5. Aggregated LTP blocks are size-limited (not time-limited).

User Input:

- Item 24: **Longest Contact Period (sec)** – this is the longest contact period measured in seconds.
- Item 25: **(32/64) bit system** – this is platform dependent parameter. Heap space is specified in number of words. For a 32-bit system, each word has 32 bits; for a 64-bit system, each word has 64 bits.
- Item 26: **Additional Margin** – adds more margin to the model per user's discretion

Model Output:

- Item 27: **Recommended HeapWord Size (with 40% for ZCO)** – this is the suggested *heapWords* value for *ionconfig*.
- Item28: **HeapWord Size (if source data no file-resident)** – this is the suggested *heapWords* value for *ionconfig* assuming the source data (1 contact worth) is copied into heap space at the time of bundle creation.

Acknowledgements

Nik Ansell co-authored/contributed to the 2016 version of this document, which has been updated and revised in 2021.