Workshop on ns-3, 2015

Implementation and Evaluation of Licklider Transmission Protocol (LTP) in ns-3

Rubén Martínez-Vidal[1], Thomas R. Henderson[2] and Joan Borrell[1]
rmartinez@deic.uab.cat

Department of Information and Communications Engineering
Universitat Autònoma de Barcelona, Spain[1]
University of Washington[2]

May 14, 2015, Castelldefels (Barcelona), Spain.

Table of Contents

- Emerging network environments pose new challenges:
  - Long round-trip times.
  - Intermittent connectivity.
  - High transmission error rates.
- When faced with these constrains standard Internet protocols:
  - Suffer severe performance degradation.
  - Become unable to operate.

New architecture and standards haven been proposed:

Delay and Disruption Tolerant Networking (DTN)

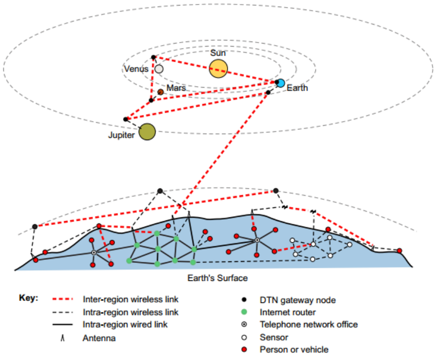
The DTN architecture:

- Is build as an **overlay network**. It can be deployed on top of both IP and non-IP based networks.
- Data is routed in a **store-and-forward** manner.
- It uses two standard protocols:
  - **Bundle protocol:**

    Encapsulates packets in a a data abstraction with variable size (bundle), adds semantic information.
  - **Licklider Transmission Protocol:**

    Specialized and optimized reliable transport protocol.

Originated from the design of the Interplanetary Internet (IPI).



Licklider Transmission Protocol:

Deep space links are often unidirectional and characterized by long delays or interruptions.

Offers reliability by means of:

- Relying on Automatic Repeat reQuest (ARQ) instead of handshakes or negotiations.
- Using coarse RTT estimation to synchronize retransmission mechanisms.
- Optimizes link usage by relying on operating system link state information
- Does not use congestion control mechanisms.

Designed to act as a reliable convergence layer for the Bundle protocol.

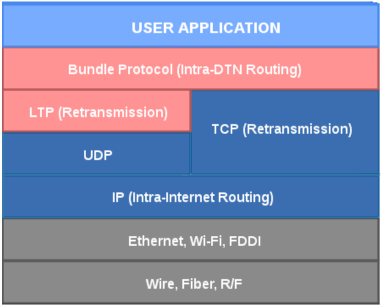
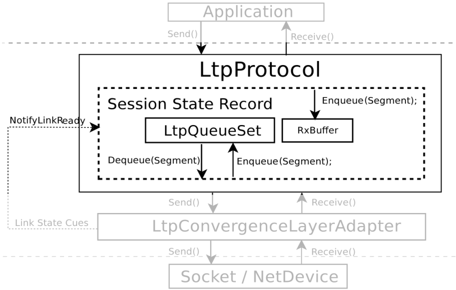


Table of Contents

Protocol Design: Basic Operation

Protocol logic and basic data structures are contained in the *LtpProtocol* class.

- The basic protocol data unit is called **segment**, its maximum size is defined by lower-layer MTU size.
- Data segment types:
  - Red Data Segments RDS: subject to retransmission.
  - Green Data Segments GDS: transmission is attempted but no guaranteed.



Protocol Design: Basic Operation

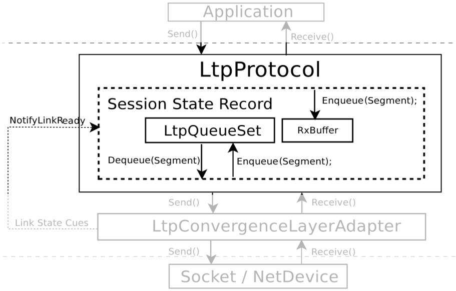
Protocol logic and basic data structures are contained in the *LtpProtocol* class.

- The final RDS is marked as end of red part (EORP) and checkpoint (CP).
- Control segments include:

  Report segment (RS): triggered by CP.

  Report ACK (RAS): triggered by RS.

  Cancellation segment (CS).
- Checkpoints and Control Segments are subject to ARQ.

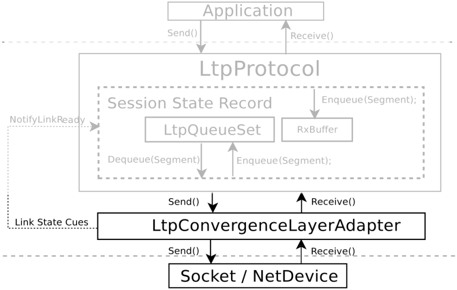


Lower-Layer Interaction

LTP may be deployed over a data-link layer protocol or over UDP. The interaction with this lower layer is offered by a Convergence Layer Adapter (CLA).

We implement the protocol over UDP and provide the **UdpConvergenceLayerAdapter** which provides the following services:

- Maps outgoing/incoming segments into send() and receive() socket operations.
- Methods to determine the MTU to avoid fragmentation.
- Link state cues (link state and tx of certain control segments).



Higher-Layer Interaction

Protocols using the LTP are referred as Client Service Instances (CSI), this role usually corresponds to Application layer services (most commonly the Bundle Protocol)

In our ns-3 implementation communication with CSI can happen in two ways:

- The CSI can make requests start/cancel transmission via invocation of methods of the *LtpProtocol* class.
- The LTP protocol issues back notifications through the use of callback functions to which the CSI subscribes beforehand.

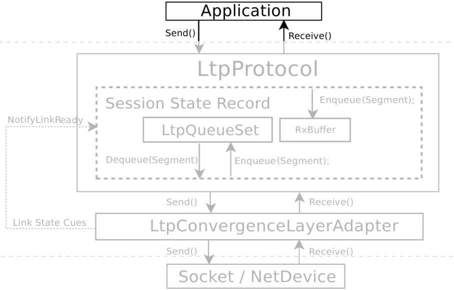


Table of Contents

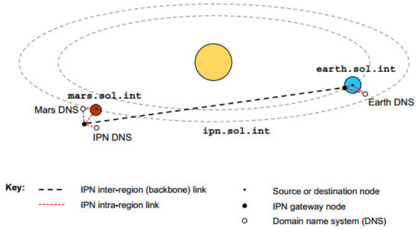
Model validation seeks to ensure the correct behavior of the protocol and verify its robustness. It was performed using two approaches:

- **Testing:** Test cases representing multiple retransmission situations.
- **Performance evaluation:** of the protocol under similar conditions to these found in the literature.

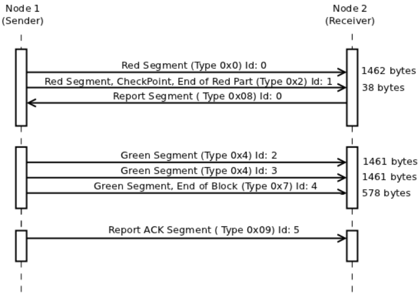
Basic transmission scenarios:

The implementation of the protocol was tested on scenarios with the following characteristics:

- Two-node topology connected with a point-to-point link using a 1500 byte MTU.
- A channel configured to emulate the characteristics of deep-space communications:
  - High channel delay.
  - ErrorModel with high error-rate.



- Transmit a 5000 bytes data block.
- Data Rate: 500kbps.
- Channel delay of 750 seconds (Earth-Mars).
- ReceiverListErrorModel to generate controlled segment losses.



We try to check all possible eventualities by performing mutiple tests combinations

- Data block containing Red and Green data.
- Losses on the Receiver or Sender side.
- Losses of data segments and control segments.
- Losses of single segment or loss of multiple segments.

Table 1: Retransmission tests using a 5000 bytes data block.

	Red Data Length	Lost packet ID (receiver)	Lost report segment (sender)	Rx Bytes
1	1500			5000
2	1500	0		5000
3	1500	1		5000
4	1500	0 , 1		5000
5	1500		✓	5000
6	1500	5	✓	5000
7	1500	2		3539
8	1500	2 , 3		2078
9	1500	0 , 3		3539
10	1500	1 , 3		3539
11	1500	0 , 1 , 2 , 3		2078
12	1500	2	✓	3539
13	1500	4	✓	4422
14	5000			5000
15	5000	0 , 1		5000
16	0			5000
17	0	1 , 2		2078

We analyze the network performance according to the **goodput** at application level. Our performance analysis tries to mimic the conditions used in real evaluations of LTP found in the literature[1].

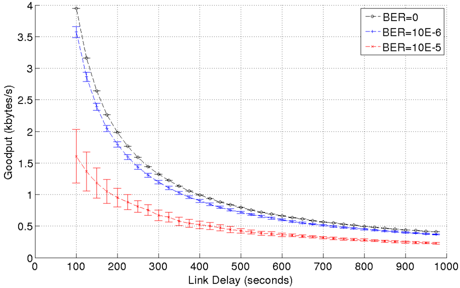
Impact of channel Bit Error Rate (BER) on LTP Performance

- Transmit 1 MB data block.
- Data rate: 115 kbps.
- Channel delay of 100 to 1000 seconds.
- RateErrorModel with varying error-rates:

  0 - error free.

  10e-6 - moderate (common space link conditions).

  10e-5 - worst case.



Points represent the mean value over a sample of 10 simulation runs.

[1] R. Wang, S. C. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," IEEE/ACM Trans. Netw., vol. 19, pp. 359–368, Apr. 2011.

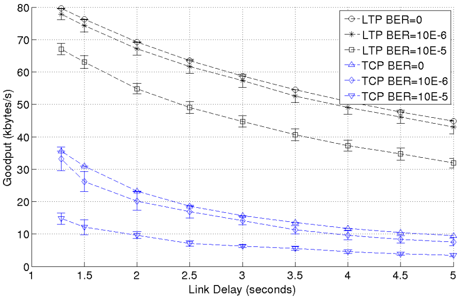
We analyze the network performance according to the **goodput** at application level. Our performance analysis tries to mimic the conditions used in real evaluations of LTP found in the literature[1].

LTP/TCP performance comparison:

- Transmit 1 MB data block.
- Data rate: 115 kbps.
- Channel delay from 1.5 to 5 seconds (Earth-Moon).
- RateErrorModel with varying error-rates: (0, 10e-6, 10e-5).

- TCP transmission rate degrades significantly.
- LTP roughly doubles the goodput of TCP.

Limitations in the currently implemented ns-3 DTN models do not allow an exact reproduction of the scenario. Generally, the results are consistent with those observed in [1].



Points represent the mean value over a sample of 10 simulation runs.

[1] R. Wang, S. C. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," IEEE/ACM Trans. Netw., vol. 19, pp. 359–368, Apr. 2011.

Table of Contents

We further validate our ns-3 LTP model for:

- **Interoperability:** It is capable of interoperating with an independent implementation and provides comparable performance.

For this validation we use the Common Open Research Emulator (CORE):

- CORE is a framework for network emulation experiments.
- Its backend coordinates the instantiation and configuration of Linux network namespace containers.
- It offers a GUI which allows configuration of network topologies and allows access to the containers at runtime.
- The links between nodes are instantiated by Linux bridges with link effects (delay and packet-loss) by the *netem* tool.

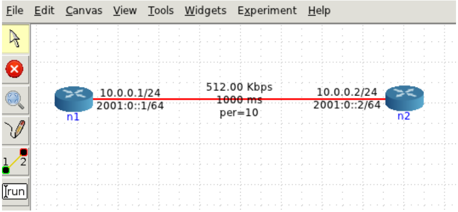
- The model is tested against the LTPlib C++ (Trinity College, Dublin) implementation.
- LTPlib operates over UDP which is the same approach used in our ns-3 model.

Validation scenario:

The network topology consists of two nodes connected by a point-to-point link.

n2 acts as the server and runs LTPLib in its associated Linux container.

n1 acts as the client and will alternatively run LTPLib or ns-3 in emulation mode.



Interoperability has been tested assuming two different working conditions:

- Error-free channel.
- Lossy channel.

We start by performing a generic test, we use both clients to perform a 1 MB file transfer and observe the packet exchanges.

Both implementations performed similarly, although a few differences can be found:

- ns-3 used full-sized IP datagrams (1500 bytes) while LTPlib typically used slightly smaller datagrams (1485 bytes).
- Spacing between segment transmissions differed, LTPlib did not send according to a regular schedule and took about 0.5 seconds longer to send the data.
- LTPlib used a different ephemeral UDP source port for each segment, while ns-3 used the same ephemeral port consistently. Both used port 1113 as the destination port.

Now we analyze the generated traffic in more detail by observing the number and type of segments generated by each implementation. Interoperability is assessed by:

- checking that the server is capable of reassembling the data.
- comparing the similarity of the segments generated by each implementation.

Obtained results:
Traffic is captured using the tcpdump utility and inspected using the Wireshark tool. In all cases the data is reassembled successfully.

Table 2: Interoperability tests: error-free channel.

Block size	Red size	LTPlib (segment type)	ns-3 (segment type)	Rx
500	500	3, RS, RAS	3, RS, RAS	✓
5000	5000	0, 0, 0, 3, RS, RAS	0, 0, 0, 3, RS, RAS	✓
500	200	3, RS, RAS	2, 7, RS, RAS	✓
5000	2000	0, 2, 4, 7, RS, RAS	0, 2, 4, 4, 7, RS, RAS	✓
500	0	7, CS	7	✓
5000	0	4,4,4,7, CS	4,4,4,7	✓

Test cases are generated by changing:

- Block size: 500 bytes and 5000 bytes (requires fragmentation).
- Transmission reliability : full-red block, full-green block, mixed block.

Slight differences:

- In the event of a mixed block the LTPlib uses fewer segments.
- In the case of a full-green block sends an additional control segment

Lastly, we show the interoperability testing over a lossy channel in which some segments may be lost.

We performed multiple tests with varying error rates (from 0 to 10%). We again used two client configurations, LTPlib and ns-3, and repeated each experiment in each configuration ten times.

PER	LTPlib client(mean/std)	ns-3 client(mean/std)
0	1.44/0.025 (sec)	0.92/0.0056 (sec)
1%	1.58/0.14 (sec)	1.03/0.11 (sec)
2%	1.61/0.12 (sec)	1.39/0.99 (sec)
5%	2.63/2.85 (sec)	1.84/1.35 (sec)
7.5%	1.78/0.11 (sec)	2.14/1.52 (sec)
10%	8.64/4.59 (sec)	3.21/2.56 (sec)

We confirmed that all transmissions eventually succeeded despite retransmissions, and then we compiled statistics on the overall data transfer delay for each trial.

- For error-free the delay for the ns-3 client was roughly half a second less (caused by the previously mentioned pacing).
- Taking this advantage int account, the two implementations perform roughly the same until the 7.5% and 10% packet error ratio cases, for which the variability in performance was generally larger.
- We observed that LTPlib was really conservative in retransmitting data, usually leading to larger latencies.

Some additional tests:

- Reversing the configuration to operate ns-3 as the LTP server confirms that ns-3 can successfully interoperate as a server,
- Robustness to mobility by performing long lasting link disconnections.

Table of Contents

Conclusion

- We offer a RFC compliant ns-3 model of LTP protocol.
- This model has been extensively tested for robustness and interoperability with an existing implementation.
- The implementation focused on offering a high fidelity model of the data packet structures, and the retransmission procedure sequence.
- The code is available at http://code.nsnam.org/rmartinez/ns-3-dev-ltp

Open Issues and Future Work:

LTP model:

- Cancellation sequence was not implemented.
- No support for concurrent transmissions.

To improve ns-3 support for DTN simulations we need to:

- Provide a fully operational DTN stack by integrating the Bundle and LTP protocols together.
- Provide models of the common baseline routing protocols used in DTNs.
- Provide sat elite models and other related mobility models.