LISP Multiplexing and Header Compression

draft-saldana-lisp-compress-mux-00

LISP Session @ IETF 96

Berlin, July 21, 2016

Jose Saldana (jsaldana@unizar.es)

Julián Fernández-Navajas (navajas@unizar.es)

José Ruiz-Mas (jruiz@unizar.es)

LISP multiplexing and Header compression

This document proposes two combined things:

- Multiplexing: send together a number of small LISP packets into a single one. They will share a single LISP header, resulting in
  - bandwidth savings (overhead reduction)
  - packet per second reduction

  It relies on **Simplemux**: draft-saldana-tsvwg-Simplemux (submitted to tsvwg).

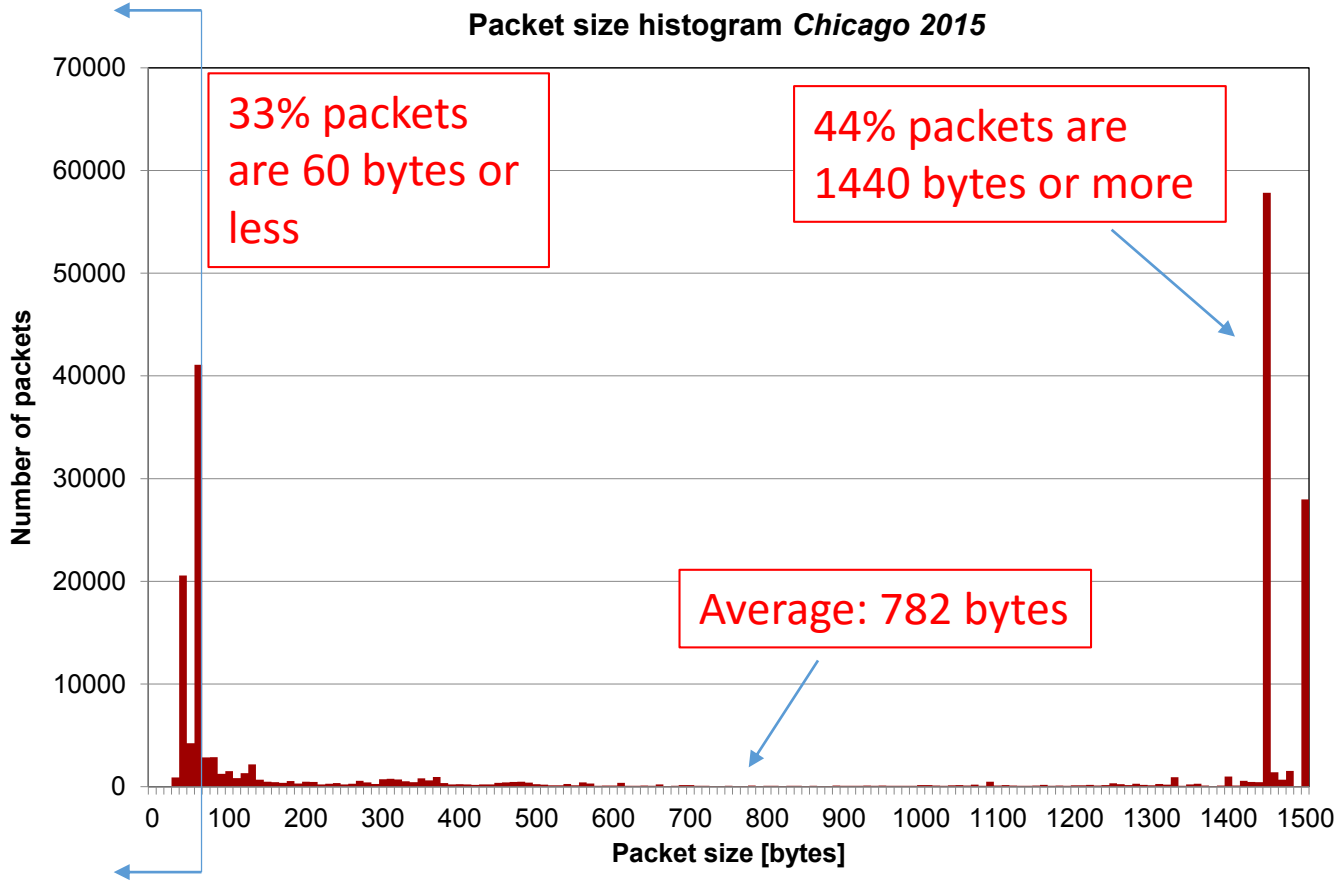
  (It adds short separators between the packets, including Protocol and Length fields. A research paper about it: *Improving Network Efficiency with Simplemux*)

- Header compression can also be applied to the EID headers.

  It relies on **ROHC**: (RObust Header Compression, RFC5795)

Why multiplexing?

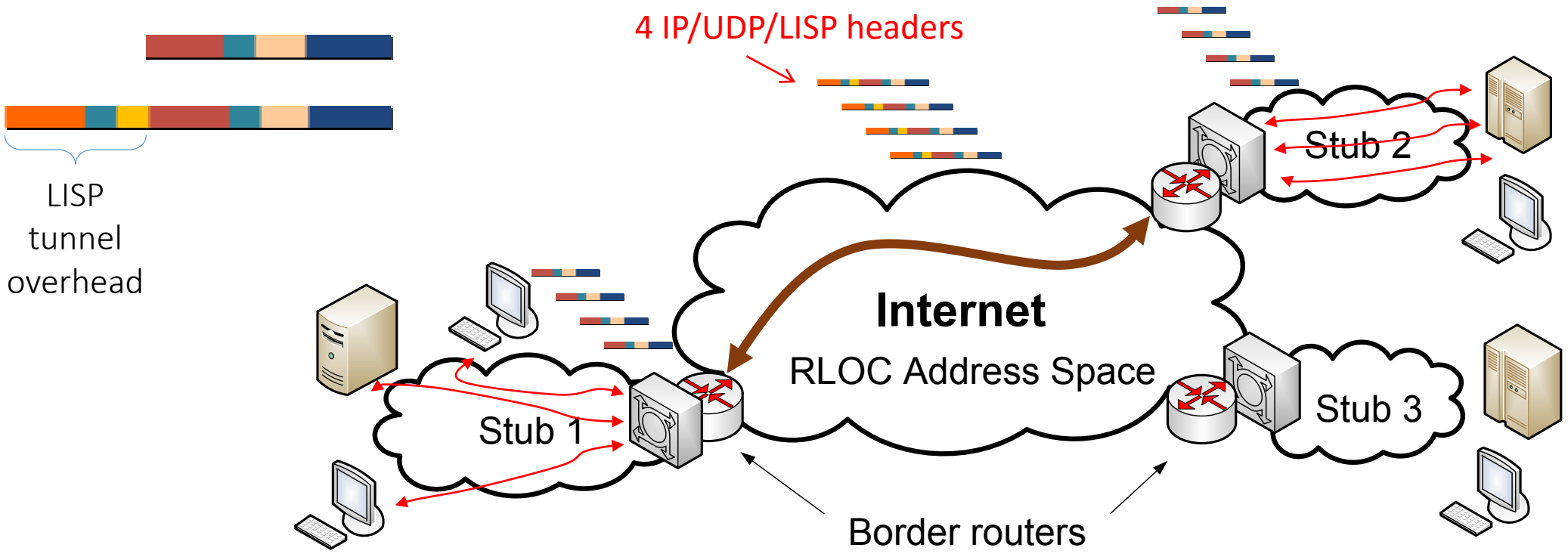
Small packets are present in the public Internet. Bad efficiency*



* Source: *https://data. caida.org/datasets/passive-2015/equinix-chicago/20150219-130000.UTC/equinix-chicago.dirA.20150219-125911.UTC.anon.pcap.gz.*
Only first 200,000 packets used

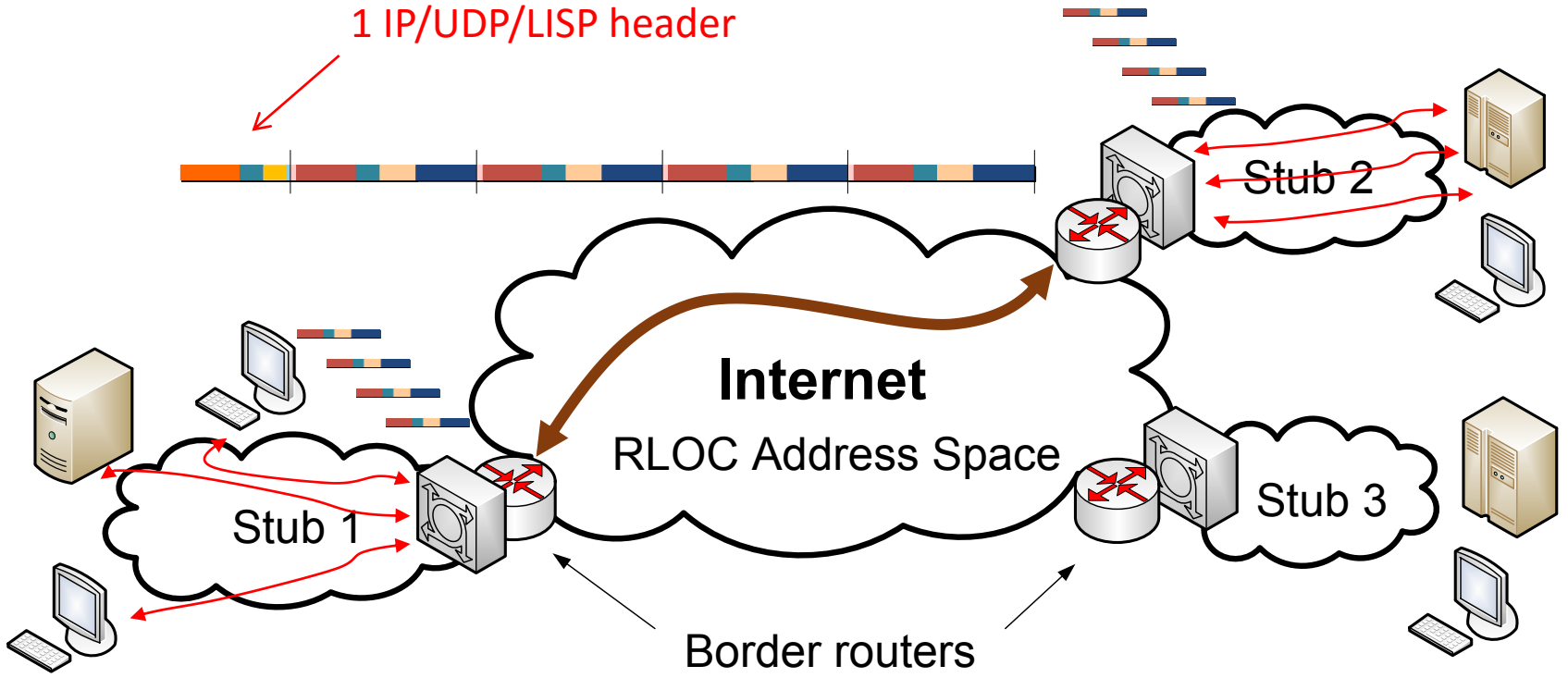
LISP multiplexing and Header compression

Packets can be grouped by the border router, in order to share the overhead of the LISP tunnel (20+8+8 bytes).



LISP multiplexing and Header compression

1 IP/UDP/LISP header can be shared by *N* packets. Improved efficiency



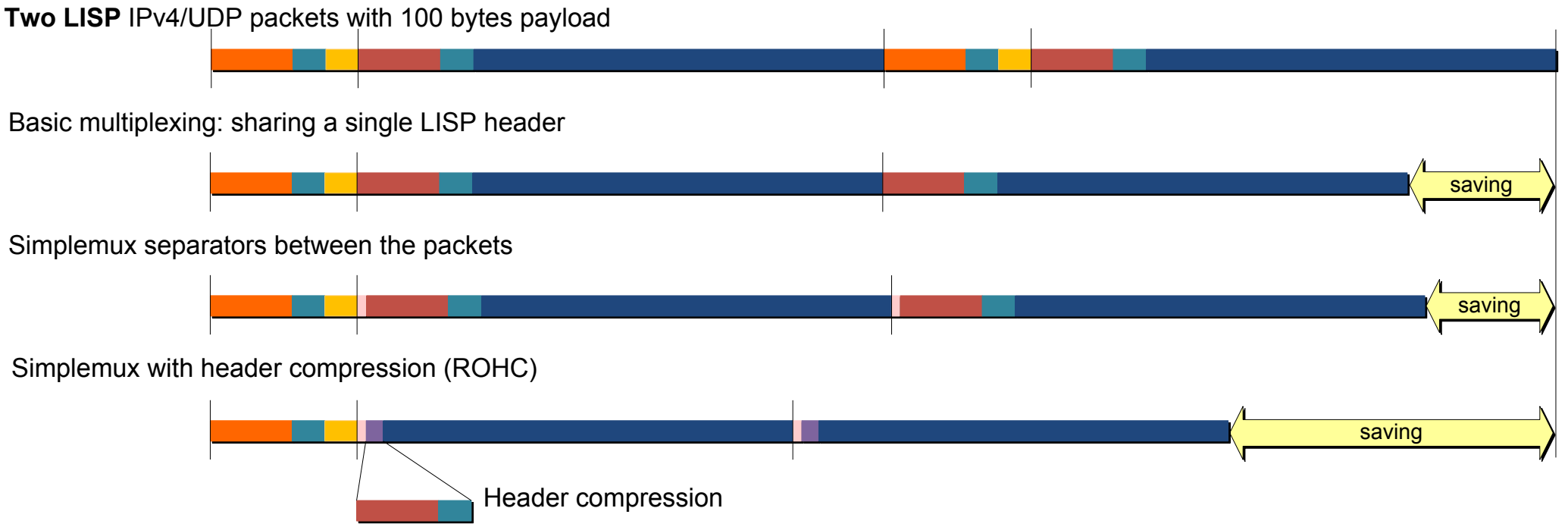
LISP multiplexing and Header compression

Three methods:

1. **Basic** multiplexing method: Just put a number of packets after the LISP header
2. Multiplexing method based on **Simplemux**: Use Simplemux separators to indicate the length of each packet
3. **Header compression & Simplemux** method: Like (2), but also using ROHC to compress the EID headers (IP, UDP or RTP)

(Real-scale examples are shown in the next slide)

Example: Two 100 byte-payload UDP packets



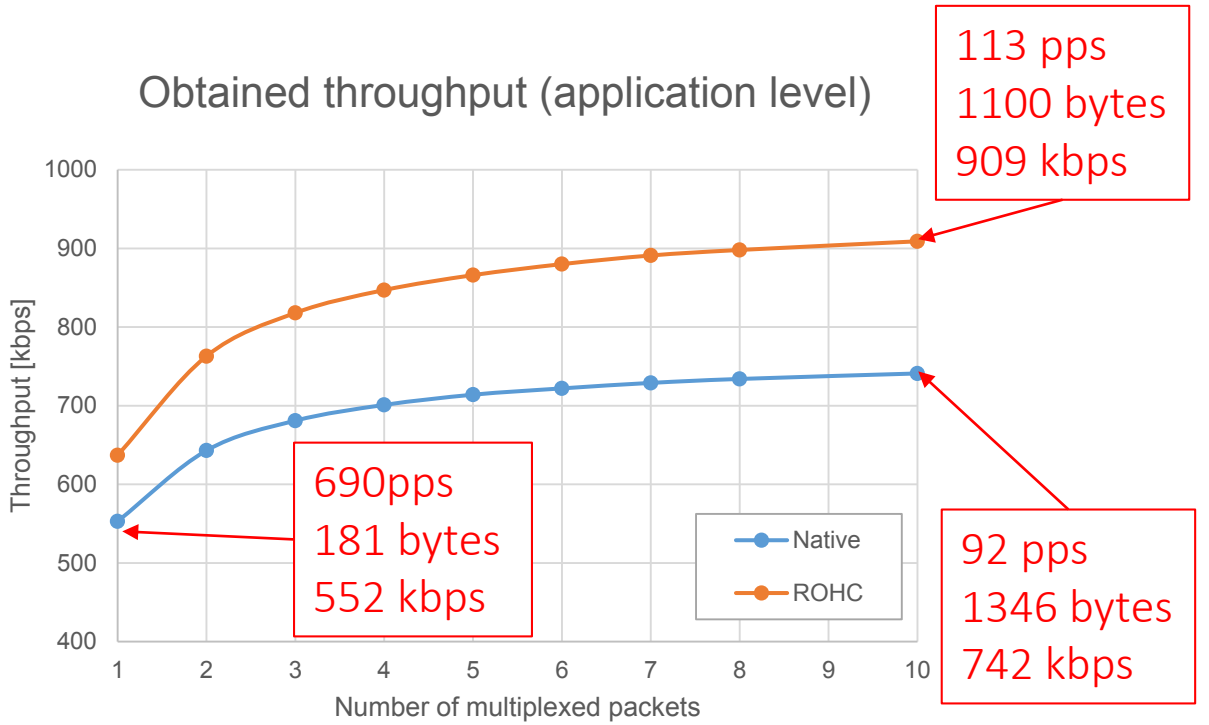
Native vs Multiplex with IPv4 over LISP

IPv4 RLOC header: 20 bytes
IPv4 EID header: 20 bytes
UDP header: 8 bytes
Simpleux header: 2 bytes
LISP header: 8 bytes
ROHC header: 4 /8 bytes
Payload

Tests with real traffic, using *iPerf* and *tc*

Traffic sent	1,5 Mbps of UDP packets with 100 bytes payload (saturated link) (128 bytes at IP level)
With LISP tunnel	128 + 36 (LISP) +3 Smux =181 bytes per packet
Traffic limit	1 Mbps at Eth level, using Linux *tc* Limit 690 pps => (x100 x8) 552 kbps at application level

Implementation based on lispmob:
https://github.com/Simplemux/lispmob-with-simplemux



We multiplex a **fixed number of packets** together.

The results shown correspond to

- method 2 (*native*): 552 to 742 kbps, 34% throughput increase
- method 3 (*ROHC*): 552 to 909 kbps, 64% throughput increase

Use cases

- A number of small packets destined to the same ETR are in the buffer
  - Small packet flows can be generated by real-time services (VoIP, games), and also by M2M or IoT traffic (e.g. samples generated by a sensor).
- **Flexibility**: Avoid dimensioning the network for the worst case. Dynamically activating multiplexing at congestion times.
  - Increasing bandwidth is a recurring cost, but adding this capability is a one-time investment.
- **Energy savings**: pps reduction with the counterpart of a short delay.
- Mobility scenarios.
  - **From the list**: *The fact that queuing is happening while waiting for a handoff to occur can be a good opportunity to pack IP packets into super-frames to send over the RAN*.
  - Multiplexing in **mobility tunnels** could make sense, as the same tunnel is used for M mobiles (a 1:M relationship). For example, in PMIPv6 (by the NETLMM WG), tunnels are created between the Local Mobility Anchor (LMA, running in the P-GW) and the Mobile Access Gateway (MAG, running in the Serving Gateway).

Backward compatibility

How can an ITR decide which encapsulation to use and which flows to multiplex while sending traffic for a particular remote EID?

- This has not yet been addressed in the draft (nor implemented).
- The LISP Canonical Address Format (LCAF) can be used.
- LISP signaling can be enriched to carry meta-information concerning whether or not to multiplex packets destined to a block of EIDs and whether or not and how to compress their headers.
  - LCAF allows effective encoding of the information concerning which traffic has to be multiplexed, based on several parameters (e.g., source and destination addresses, ToS, application, etc.).
- **From the list**: The “Encapsulation Format” LCAF type can be used to tell the ITR, on a lookup, what the ETR is willing to accept. We COULD treat this new capability as a different encapsulation type.

Thanks a lot!
Questions

Jose Saldana (jsaldana@unizar.es)
Julián Fernández-Navajas (navajas@unizar.es)
José Ruiz-Mas (jruiz@unizar.es)
University of Zaragoza

This work has been partially financed by the EU H2020 Wi-5 project (G.A. no: 644262), and the Spanish Ministry of Economy and Competitiveness project TIN2015-64770-R, in cooperation with the European Regional Development Fund.