Locator/ID Separation Protocol Working Group J. Saldana
Internet-Draft J. Fernandez Navajas
Intended status: Experimental J. Ruiz Mas
Expires: September 3, 2018 University of Zaragoza
March 2, 2018

Header compression and multiplexing in LISP
draft-saldana-lisp-compress-mux-04

Abstract

When small payloads are transmitted through a packet-switched network, the resulting overhead may result significant. This is stressed in the case of LISP, where a number of headers have to be added to each packet.

This document proposes a way to send together, into a single packet, a number of small packets, which are in the buffer of a ITR, having the same ETR as destination. This way, they can share a single LISP header, and therefore bandwidth savings can be obtained, and a reduction in the overall number of packets sent to the network can be achieved.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The rate of small packets present in the Internet is significant [Simplemux_CIT]. First, TCP Acknowledgements (ACKs), which may have no payload, are sent in every TCP connection. In addition some services with real-time and interactivity constraints (VoIP, videoconferencing, telemedicine, video surveillance, online gaming, etc.) generate a traffic profile consisting of high rates of small packets, which are necessary in order to transmit frequent updates between the two extremes of the communication. In addition, some other services also use small packets as e.g., instant messaging, M2M (Machine to Machine) services sending collected data in sensor networks or IoT scenarios using wireless links.

When small payloads are transmitted through a packet-switched network, the resulting overhead may result significant. This is more signifcant in the case of tunneling protocols, where a number of headers are prepended to a packet.

In the case of LISP, this overhead may be further stressed. As an example, an IPv4 TCP ACK (40 bytes), sent with standard LISP over IPv4 requires 76 bytes (96 if IPv6 is used by one of the IP headers). Or an RTP packet with e.g. 20 bytes of payload, using standard LISP over IPv4, requires 96 bytes (116 if IPv6 is used in one of the IP headers).

Some methods have been proposed in order to reduce LISP’s overhead, with the aim of avoiding MTU issues, as e.g. [I-D.boucadair-lisp-v6-compact-header].

When a number of small packets are in the buffer of an ITR, having the same ETR as destination, they can be sent together, sharing a single LISP header, and simultaneously obtaining three benefits: a) bandwidth savings; b) a reduction in the number of packets, which may also be translated into c) a reduction of the overall energy consumption of network equipment. According to [Efficiency] internal packet processing engines and switching fabric require 60% and 18% of the power consumption of high-end routers respectively. Thus, reducing the number of packets to be managed will reduce the overall energy consumption. The measurements deployed in [Power] on commercial routers corroborate this fact: a study using different packet sizes was presented, and the tests with big packets showed a reduction of the energy consumption, since a certain amount of energy is associated to header processing tasks, and not only to the sending of the packet itself.

All in all, another trade-off appears: on the one hand, energy consumption is increased in the two extremes due to header compression processing; on the other hand, energy consumption is reduced in the intermediate nodes because of the reduction of the number of packets transmitted. This tradeoff should be explored more deeply.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Native LISP and proposed solutions

A LISP encapsulated packet, as defined in [RFC6830], has the next structure (Figure 1):

```
+--+---+----+--+---+-------+
|OH|UDP|LISP|IH|TrH|payload|
+--+---+----+--+---+-------+
|            |              |
<---LISP----><-----pkt----->
```

Figure 1: Structure of a LISP encapsulated packet

Where each of the headers corresponds to:

- OH: The outer header containing RLOCs obtained from the ingress router's EID-to-RLOC Cache.

- UDP Header, as required by [RFC6830]. The destination port MUST be set to the IANA-assigned port value 4341.

- LISP-specific 8-octet header.

- IH is the Inner Header on the datagram received from the originating host. The source and destination IP addresses are EIDs.

- TrH: The Transport Header, i.e. a TCP, UDP or SCTP header.

Note that [RFC6830] defines "LISP Header" as a set including:

- the outer IPv4 or IPv6 header;

- a UDP header;

- a LISP-specific 8-octet header that follows the UDP header.

2.1. Basic multiplexing method

When a number of small packets (e.g. VoIP, TCP ACKs, etc.) are stored in the output buffer of an ITR, it MAY be possible to send a number of them into a single RLOC-space packet, thus reducing the overhead and the number of packets at the same time. This may have some additional benefits, as the reduction of the amount of packets travelling between the ITR and the ETR. It may also result in a reduction of the processing requirements in intermediate nodes, which may be transalted into certain energy savings.

A very strightforward solution for multiplexing a number of EID-space packets into a single RLOC-space one is to just concatenate a number of IP packets after the LISP Header (see Figure 2).

One of the free bits in the LISP header should be used to flag the fact that more than a single packet is included in the encapsulated one.

```
+--+---+----+--+---+-------+--+---+-------+--+---+-------+
|OH|UDP|LISP|IH|TrH|payload|IH|TrH|payload|IH|TrH|payload|
+--+---+----+--+---+-------+--+---+-------+--+---+-------+
|           |              |              |              |
<---LISP----><---pkt #1----><----pkt #2---><----pkt #3--->
```

Figure 2: Structure of a LISP packet encapsulating three IP packets

When an ETR receives a packet with the indication that it contains more than a single packet (this is achieved by using a port number different from 4341 in the UDP header preceding the LISP header), it first extracts all the content after the LISP header, and then it uses the "Total Length" field of the Inner IP Header to know the length of the first packet. Once extracted, it removes the packet and assumes the next bytes correspond to the next IP Header, so it can subsequently extract all the included packets.

2.2. Multiplexing method based on Simplemux

Simplemux [I-D.saldana-tsvwg-simplemux] is a simple multiplexing protocol that allows the inclusion of a whole packet belonging to any protocol ("tunneled packet") into any tunneling protocol. It includes a Lenght field, expressing the length of the multiplexed packet, and a Protocol field, expressing the protocol to which the tunneled packet belongs.

If a Simplemux separator is placed after the LISP header, then a number of packets can be included, taking into account that the Simplemux separator includes a field expressing the length of the next packet. In the present case, LISP is used as the tunneling protocol.

A port number different from 4341 should be used in the UDP header preceding the LISP header, in order to indicate that the protocol inside the LISP header is not IP but Simplemux.

```
+--+---+----+----+--+---+-------+----+--+---+-------+----+--+---+-------+
|OH|UDP|LISP|Smux|IH|TrH|payload|Smux|IH|TrH|payload|Smux|IH|TrH|payload|
+--+---+----+----+--+---+-------+----+--+---+-------+----+--+---+-------+
|           |                   |                   |                   |
<---LISP----><------pkt #1------><------pkt #2------><------pkt #3------>
```

Figure 3: Structure of a LISP packet encapsulating three IP packets separated with Simplemux

2.3. Header compression and multiplexing method

ROHC (RObust Header Compression [RFC5795]) is able to compress UDP/IP, ESP/IP and RTP/UDP/IP headers. It is a robust scheme developed for header compression over links with high bit error rates, such as wireless ones. It incorporates mechanisms for quick resynchronization of the context, with an improved encoding scheme for compressing the header fields that change.

Taking into account that the inner packets are tunneled with LISP, header compression can be used in order to remove those fields that are the same for every packet in a flow.

The "Protocol" field of Simplemux allows the possibility of indicating that the packets are compressed with ROHC [RFC5795]. The protocol number 142 is used for this, as defined in [RFC5858].

```
+--+---+----+----+----+-------+----+----+-------+----+----+-------+
|OH|UDP|LISP|Smux|RoHC|payload|Smux|RoHC|payload|Smux|RoHC|payload|
+--+---+----+----+----+-------+----+----+-------+----+----+-------+
|           |                 |                 |                 |
<---LISP----><-----pkt #1-----><-----pkt #2-----><-----pkt #3----->
```

Figure 4: Structure of a LISP packet encapsulating three packets compressed with ROHC separated with Simplemux

3. Acknowledgements

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

4. IANA Considerations

The present document proposes the use of a Simplemux separator after the LISP header, so a port number different from 4341 should be used in the UDP header preceding the LISP header.

5. Security Considerations

The mechanism proposed in the present draft has been developed in such a way that packet aggregation and security can be simultaneously applied to the same traffic flows, i.e. a single security header could protect a number of packets belonging to different flows.

As a consequence, the overall efficiency could be improved, as the number of security headers could be reduced from N (being N the number of multiplexed packets) to 1.

The proposed mechanism could work in cooperation with LISP-Security [I-D.ietf-lisp-sec].

6. References

6.1. 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>.

[RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <https://www.rfc-editor.org/info/rfc5795>.

[RFC5858] Ertekin, E., Christou, C., and C. Bormann, "IPsec Extensions to Support Robust Header Compression over IPsec", RFC 5858, DOI 10.17487/RFC5858, May 2010, <https://www.rfc-editor.org/info/rfc5858>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

6.2. Informative References

[Efficiency] Bolla, R., Bruschi, R., Davoli, F., and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures", IEEE Communications Surveys and Tutorials vol.13, no.2, pp.223,244, 2011.

[I-D.boucadair-lisp-v6-compact-header] Boucadair, M. and C. Jacquenet, "A Compact LISP Encapsulation Scheme to Transport IPv4 Packets over an IPv6 Network", draft-boucadair-lisp-v6-compact-header-04 (work in progress), Dec 2016.

[I-D.ietf-lisp-sec] Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-14 (work in progress), October 2017.

[I-D.saldana-tsvwg-simplemux] Saldana, J., "Simplemux. A generic multiplexing protocol", draft-saldana-tsvwg-simplemux-06 (work in progress), January 2017.

[Power] Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and S. Wright, "Power Awareness in Network Design and Routing", INFOCOM 2008. The 27th Conference on Computer Communications. IEEE pp.457,465, 2008.

[Simplemux_CIT]
Saldana, J., Forcen, I., Fernandez-Navajas, J., and J. Ruiz-Mas, "Improving Network Efficiency with Simplemux", IEEE CIT 2015, International Conference on Computer and Information Technology , pp. 446-453, 26-28 October 2015, Liverpool, UK, 2015.

Authors' Addresses

Jose Saldana
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza 50018
Spain

Phone: +34 976 762 698
Email: jsaldana@unizar.es

Julian Fernandez Navajas
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza 50018
Spain

Phone: +34 976 761 963
Email: navajas@unizar.es

Jose Ruiz Mas
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza 50018
Spain

Phone: +34 976 762 158
Email: jruiz@unizar.es