Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 3, 2016

X. Xu
Huawei Technologies
R. Asati
Cisco Systems
T. Herbert
Facebook
L. Yong
Huawei USA
Y. Lee
Comcast
Y. Fan
China Telecom
I. Beijnum
Institute IMDEA Networks

January 31, 2016

Encapsulating IP in UDP
draft-xu-intarea-ip-in-udp-03

Abstract

Existing Softwire encapsulation technologies are not adequate for efficient load balancing of Softwire service traffic across IP networks. This document specifies additional Softwire encapsulation technology, referred to as IP-in-UDP (User Datagram Protocol), which can facilitate the load balancing of Softwire service traffic across IP networks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

To fully utilize the bandwidth available in IP networks and/or facilitate recovery from a link or node failure, load balancing of traffic over Equal Cost Multi-Path (ECMP) and/or Link Aggregation Group (LAG) across IP networks is widely used. [RFC5640] describes a method for improving the load balancing efficiency in a network carrying Softwire Mesh service [RFC5565] over Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] and Generic Routing Encapsulation (GRE) [RFC2784] encapsulations. However, this method requires core routers to perform hash calculation on the "load-balancing" field contained in tunnel encapsulation headers (i.e., the Session ID field in L2TPv3 headers or the Key field in GRE headers), which is not widely supported by existing core routers.

Most existing routers in IP networks are already capable of distributing IP traffic "microflows" [RFC2474] over ECMP paths and/or

LAG based on the hash of the five-tuple of User Datagram Protocol (UDP) [RFC0768] and Transmission Control Protocol (TCP) packets (i.e., source IP address, destination IP address, source port, destination port, and protocol). By encapsulating the Softwire service traffic into an UDP tunnel and using the source port of the UDP header as an entropy field, the existing load-balancing capability as mentioned above can be leveraged to provide fine-grained load-balancing of Softwire service traffic traffic over IP networks. This is similar to why LISP [RFC6830] , MPLS-in-UDP [RFC7510] and VXLAN [RFC7348] use UDP encapsulation. Therefore, this specification defines an IP-in-UDP encapsulation method dedicated for Softwire service (including both mesh and hub-spoke modes).

IPv6 flow label has been proposed as an entropy field for load balancing in IPv6 network environment [RFC6438]. However, as stated in [RFC6936], the end-to-end use of flow labels for load balancing is a long-term solution and therefore the use of load balancing using the transport header fields would continue until any widespread deployment is finally achieved. As such, IP-in-UDP encapsulation would still have a practical application value in the IPv6 networks during this transition timeframe.

Similarly, the IP-in-UDP encapsulation format defined in this document by itself cannot ensure the integrity and privacy of data packets being transported through the IP-in-UDP tunnels and cannot enable the tunnel decapsulators to authenticate the tunnel encapsulator. Therefore, in the case where any of the above security issues is concerned, the IP-in-UDP SHOULD be secured with IPsec [RFC4301] or DTLS [RFC6347]. For more details, please see Section 6 of Security Considerations.

1.1. Conventions

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. Terminology

This memo makes use of the terms defined in [RFC5565].

3. Encapsulation in UDP

IP-in-UDP encapsulation format is shown as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Source Port = Entropy      |         Dest Port = TBD1      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           UDP Length          |         UDP Checksum          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                           IP Packet                           ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Source Port of UDP

This field contains a 16-bit entropy value that is generated by the encapsulator to uniquely identify a flow. What constitutes a flow is locally determined by the encapsulator and therefore is outside the scope of this document. What algorithm is actually used by the encapsulator to generate an entropy value is outside the scope of this document.

In case the tunnel does not need entropy, this field of all packets belonging to a given flow SHOULD be set to a randomly selected constant value so as to avoid packet reordering.

To ensure that the source port number is always in the range 49152 to 65535 (Note that those ports less than 49152 are reserved by IANA to identify specific applications/protocols) which may be required in some cases, instead of calculating a 16-bit hash, the encapsulator SHOULD calculate a 14-bit hash and use those 14 bits as the least significant bits of the source port field while the most significant two bits SHOULD be set to binary 11. That still conveys 14 bits of entropy information which would be enough as well in practice.

Destination Port of UDP

This field is set to a value (TBD1) allocated by IANA to indicate that the UDP tunnel payload is an IP packet. As for whether the encapsulated IP packet is IPv4 or IPv6, it would be determined according to the Version field in the IP header of the encapsulated IP packet.

UDP Length

The usage of this field is in accordance with the current UDP specification [RFC0768].

UDP Checksum

For IPv4 UDP encapsulation, this field is RECOMMENDED to be set to zero for performance or implementation reasons because the IPv4 header includes a checksum and use of the UDP checksum is optional with IPv4. For IPv6 UDP encapsulation, the IPv6 header does not include a checksum, so this field MUST contain a UDP checksum that MUST be used as specified in [RFC0768] and [RFC2460] unless one of the exceptions that allows use of UDP zero-checksum mode (as specified in [RFC6935]) applies.

IP Packet

This field contains one IP packet.

4. Processing Procedures

This IP-in-UDP encapsulation causes E-IP[RFC5565] packets to be forwarded across an I-IP [RFC5565] transit core via "UDP tunnels". While performing IP-in-UDP encapsulation, an ingress AFBR (e.g. PE router) would generate an entropy value and encode it in the Source Port field of the UDP header. The Destination Port field is set to a value (TBD1) allocated by IANA to indicate that the UDP tunnel payload is an IP packet. Transit routers, upon receiving these UDP encapsulated IP packets, could balance these packets based on the hash of the five-tuple of UDP packets. Egress AFBRs receiving these UDP encapsulated IP packets MUST decapsulate these packets by removing the UDP header and then forward them accordingly (assuming that the Destination Port was set to the reserved value pertaining to IP).

Similar to all other Softwire tunneling technologies, IP-in-UDP encapsualtion introduces overheads and reduces the effective Maximum Transmision Unit (MTU) size. IP-in-UDP encapsulation may also impact Time-to-Live (TTL) or Hop Count (HC) and Differentiated Services (DSCP). Hence, IP-in-UDP MUST follow the corresponding procedures defined in [RFC2003].

Ingress AFBRs MUST NOT fragment I-IP packets (i.e., UDP encapsulated IP packets), and when the outer IP header is IPv4, ingress AFBRs MUST set the DF bit in the outer IPv4 header. It is strongly RECOMMENDED that I-IP transit core be configured to carry an MTU at least large enough to accommodate the added encapsulation headers. Meanwhile, it is strongly RECOMMENDED that Path MTU Discovery [RFC1191] [RFC1981] is used to prevent or minimize fragmentation. Once an ingress AFBR needs to perform fragmentation on an E-IP packet before encapsulating, it MUST use the same source UDP port for all fragmented packets so as to ensures these fragmented packets are

always forwarded on the same path. In a word, IP-in-UDP is just applicable in those Softwire network environments where fragmentation on the tunnel layer is not needed.

5. Congestion Considerations

Section 3.1.3 of [RFC5405] discussed the congestion implications of UDP tunnels. As discussed in [RFC5405], because other flows can share the path with one or more UDP tunnels, congestion control [RFC2914] needs to be considered. As specified in [RFC5405]:

> "IP-based traffic is generally assumed to be congestion-controlled, i.e., it is assumed that the transport protocols generating IP-based traffic at the sender already employ mechanisms that are sufficient to address congestion on the path. Consequently, a tunnel carrying IP-based traffic should already interact appropriately with other traffic sharing the path, and specific congestion control mechanisms for the tunnel are not necessary".

Since IP-in-UDP is only used to carry IP traffic which is generally assumed to be congestion controlled by the transport layer, it generally does not need additional congestion control mechanisms.

6. Security Considerations

The security problems faced with the IP-in-UDP tunnel are exactly the same as those faced with IP-in-IP [RFC2003] and IP-in-GRE tunnels [RFC2784]. In other words, the IP-in-UDP tunnel as defined in this document by itself cannot ensure the integrity and privacy of data packets being transported through the IP-in-UDP tunnel and cannot enable the tunnel decapsulator to authenticate the tunnel encapsulator. In the case where any of the above security issues is concerned, the IP-in-UDP tunnel SHOULD be secured with IPsec or DTLS. IPsec was designed as a network security mechanism and therefore it resides at the network layer. As such, if the tunnel is secured with IPsec, the UDP header would not be visible to intermediate routers anymore in either IPsec tunnel or transport mode. As a result, the meaning of adopting the IP-in-UDP tunnel as an alternative to the IP-in-GRE or IP-in-IP tunnel is lost. By comparison, DTLS is better suited for application security and can better preserve network and transport layer protocol information. Specifically, if DTLS is used, the destination port of the UDP header will be filled with a value (TBD2) indicating IP with DTLS and the source port can still be used as an entropy field for load-sharing purposes.

If the tunnel is not secured with IPsec or DTLS, some other method should be used to ensure that packets are decapsulated and forwarded

by the tunnel tail only if those packets were encapsulated by the tunnel head. If the tunnel lies entirely within a single administrative domain, address filtering at the boundaries can be used to ensure that no packet with the IP source address of a tunnel endpoint or with the IP destination address of a tunnel endpoint can enter the domain from outside. However, when the tunnel head and the tunnel tail are not in the same administrative domain, this may become difficult, and filtering based on the destination address can even become impossible if the packets must traverse the public Internet. Sometimes only source address filtering (but not destination address filtering) is done at the boundaries of an administrative domain. If this is the case, the filtering does not provide effective protection at all unless the decapsulator of an IP-in-UDP validates the IP source address of the packet.

7. IANA Considerations

One UDP destination port number indicating IP needs to be allocated by IANA:

Service Name: IP-in-UDP

Transport Protocol(s): UDP

Assignee: IESG <iesg@ietf.org>

Contact: IETF Chair <chair@ietf.org>.

Description: Encapsulate IP packets in UDP tunnels.

Reference: This document.

Port Number: TBD1 -- To be assigned by IANA.

One UDP destination port number indicating IP with DTLS needs to be allocated by IANA:

Service Name: IP-in-UDP-with-DTLS

Transport Protocol(s): UDP

Assignee: IESG <iesg@ietf.org>

Contact: IETF Chair <chair@ietf.org>.

Description: Encapsulate IP packets in UDP tunnels with DTLS.

Reference: This document.

Port Number: TBD2 -- To be assigned by IANA.

8. Acknowledgements

Thanks to Vivek Kumar, Carlos Pignataro and Mark Townsley for their valuable comments on the initial idea of this document. Thanks to Andrew G. Malis for their valuable comments on this document.

9. References

9.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <http://www.rfc-editor.org/info/rfc768>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <http://www.rfc-editor.org/info/rfc1191>.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, DOI 10.17487/RFC1981, August 1996, <http://www.rfc-editor.org/info/rfc1981>.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <http://www.rfc-editor.org/info/rfc2003>.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <http://www.rfc-editor.org/info/rfc2784>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <http://www.rfc-editor.org/info/rfc4301>.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, DOI 10.17487/RFC5405, November 2008, <http://www.rfc-editor.org/info/rfc5405>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <http://www.rfc-editor.org/info/rfc6935>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <http://www.rfc-editor.org/info/rfc6936>.

9.2. Informative References

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <http://www.rfc-editor.org/info/rfc2474>.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <http://www.rfc-editor.org/info/rfc2914>.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <http://www.rfc-editor.org/info/rfc3931>.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, DOI 10.17487/RFC5565, June 2009, <http://www.rfc-editor.org/info/rfc5565>.

[RFC5640] Filsfils, C., Mohapatra, P., and C. Pignataro, "Load-Balancing for Mesh Softwires", RFC 5640, DOI 10.17487/RFC5640, August 2009, <http://www.rfc-editor.org/info/rfc5640>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <http://www.rfc-editor.org/info/rfc6438>.

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

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

Authors' Addresses

Xiaohu Xu
Huawei Technologies
No.156 Beiqing Rd
Beijing 100095
CHINA

Phone: +86-10-60610041
Email: xuxiaohu@huawei.com

Rajiv Asati
Cisco Systems
7200 Kit Creek Road
Research Triangle Park,, NC 27709
USA

Email: rajiva@cisco.com

Tom Herbert
Facebook
1 Hacker Way,
Menlo Park, CA 94052
USA

Email: tom@herbertland.com

Lucy Yong
Huawei USA
5340 Legacy Dr
Plano, TX 75025
USA

Email: Lucy.yong@huawei.com

Yiu Lee
Comcast
One Comcast Center
Philadelphia, PA
USA

Email: Yiu_Lee@Cable.Comcast.com

Yongbing Fan
China Telecom
Guangzhou
CHINA

Email: fanyb@gsta.com

Iljitsch van Beijnum
Institute IMDEA Networks
Avda. del Mar Mediterraneo, 22
Leganes,, Madrid 28918
Spain

Email: iljitsch@muada.com