INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: April 2016

T. Herbert
Facebook
F. Templin
Boeing Research & Technology

October 19, 2015

Fragmentation option for Generic UDP Encapsulation
draft-herbert-gue-fragmentation-02

Abstract

This specification describes a fragmentation and reassembly capability with an associated header option for Generic UDP Encapsulation.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

This specification describes a fragmentation and reassembly capability in Generic UDP Encapsulation (GUE) [I.D.ietf-nvo3-gue]. This entails adding a GUE option and procedures for fragmentation and reassembly in the encapsulation layer. This specification adapts the procedures for IP fragmentation and reassembly described in [RFC0791] and [RFC2460]. Fragmentation may be performed on both data and control messages in GUE.

1.1 Motivation

This section describes the motivation for having a fragmentation option in GUE.

MTU and fragmentation issues with In-the-Network Tunneling are described in [RFC4459]. Considerations need to be made when a packet is received at a tunnel ingress point which may be too large to traverse the path between tunnel endpoints.

There are four suggested alternatives in [RFC4459] to deal with this:

1) Fragmentation and Reassembly by the Tunnel Endpoints

2) Signaling the Lower MTU to the Sources

3) Encapsulate Only When There is Free MTU

4) Fragmentation of the Inner Packet

Many tunneling protocol implementations have assumed that fragmentation should be avoided, and in particular alternative #3 seems preferred for deployment. In this case, it is assumed that an operator can configure the MTUs of links in the paths of tunnels to ensure that they are large enough to accommodate any packets and required encapsulation overhead. This method, however, may not be feasible in certain deployments and may be prone to misconfiguration in others.

Similarly, the other alternatives have drawbacks that are described in [RFC4459]. Alternative #2 implies use of something like Path MTU Discovery which is not known to be sufficiently reliable. Alternative #4 is not permissible with IPv6 or when the DF bit is set for IPv4, and it also introduces other known issues with IP fragmentation.

For alternative #1, fragmentation and reassembly at the tunnel endpoints, there are two possibilities: encapsulate the large packet and then perform IP fragmentation, or segment the packet and then

encapsulate each segment (a non-IP fragmentation approach).

Performing IP fragmentation on an encapsulated packet has the same issues as that of normal IP fragmentation. Most significant of these is that the Identification field is only sixteen bits in IPv4 which introduces problems with wraparound as described in [FRAGHRM].

The second possibility follows the suggestion expressed in [RFC2764] and the fragmentation feature described in the AERO protocol [I.D.templin-aerolink], that is for the tunneling protocol itself to incorporate a segmentation and reassembly capability that operates at the tunnel level. In this method fragmentation is part of the encapsulation and an encapsulation header contains the information for reassembly. This is different from IP fragmentation in that the IP headers of the original packet are not replicated for each fragment.

Incorporating fragmentation into the encapsulation protocol has some advantages:

- A 32 bit identifier can be defined to avoid issues of the 16 bit Identification in IPv4.
- Encapsulation mechanisms for security and identification such as virtual network identifiers can be applied to each segment.
- This allows the possibility of using alternate fragmentation and reassembly algorithms (e.g. fragmentation with Forward Error Correction).
- Fragmentation is transparent to the underlying network so it is unlikely that fragmented packet will be unconditionally dropped as might happen with IP fragmentation.

1.2 Scope

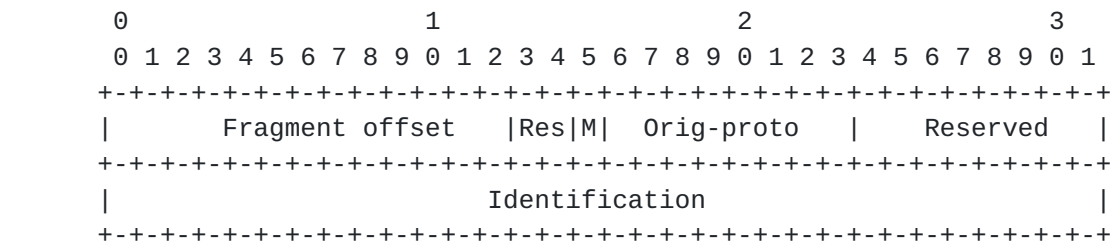
This specification describes the mechanics of fragmentation in Generic UDP Encapsulation. The operational aspects and details for higher layer implementation must be considered for deployment, but are considered out of scope for this document. The AERO protocol [I.D.templin-aerolink] defines one use case of fragmentation with encapsulation.

1.3 Terminology

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 Option format

Fragments in GUE are sent with a fragmentation option in the GUE header. The format of this option is:



- Fragment offset: This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

- Res: Two bit reserved field. Must be set to zero for transmission. If set to non-zero in a received packet then the packet MUST be dropped.

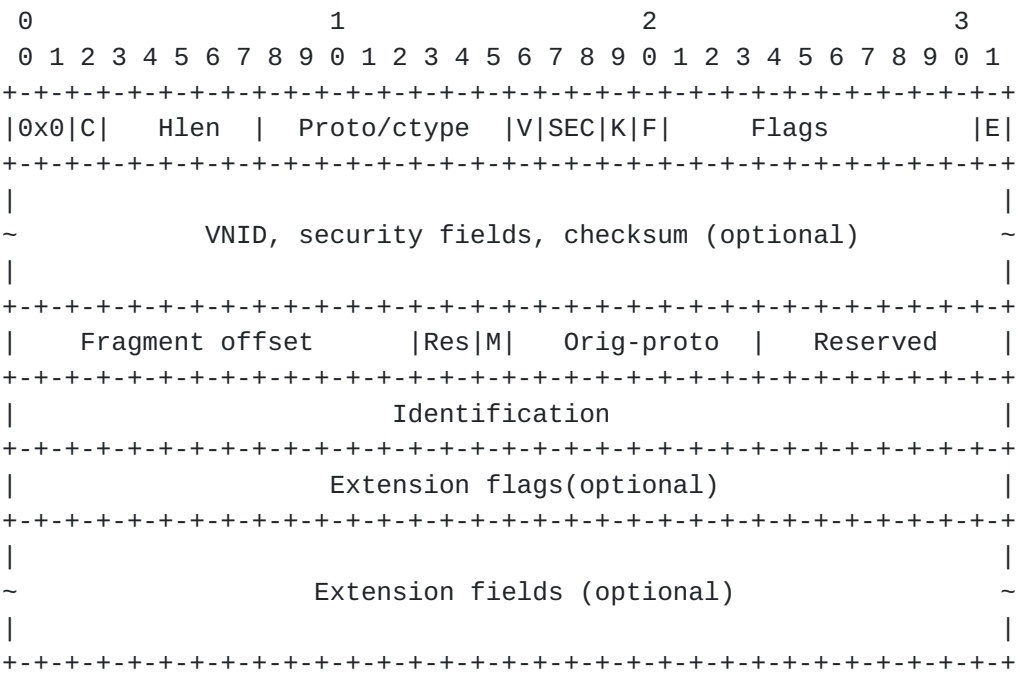
- M: More fragments bit. Set to 1 when there are more fragments following in the datagram, set to 0 for the last fragment.

- Orig-proto: The control type (when C bit is set) or the IP protocol (when C bit is not set) of the fragmented packet.

- Reserved: Must be set to 0 on transmission. If set to non-zero in a received packet then the packet MUST be dropped.

- Identification: Identifies fragments of a fragmented packet.

The format of the fragmentation option within the GUE header is:



Pertinent fields to fragmentation are:

- C: This bit is set for each fragment based on the whether the original packet being fragmented is a control or data message.
- Proto/ctype - For the first fragment (fragment offset is zero) this is set to that of the original packet being fragmented (either will be a control type or IP protocol). For other fragments, this is set to zero for a control message being fragmented, or to "No next header" (protocol number 59) for a data message being fragmented.
- F bit - Set to indicate presence of the fragmentation option field.

3 Procedures

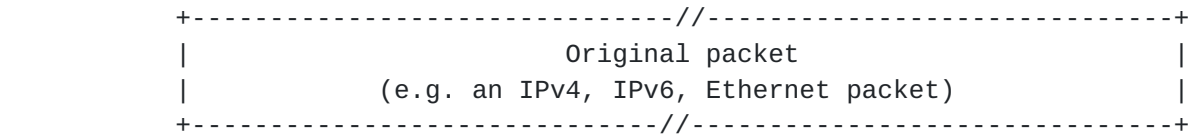
3.1 Fragmentation

If an encapsulator determines that a packet must be fragmented (eg. the packets size exceed the Path MTU of the tunnel) it may divide the packet into fragments and send each fragment as a separate GUE packet, to be reassembled at the decapsulator (tunnel egress).

For every packet that is to be fragmented, the source node generates

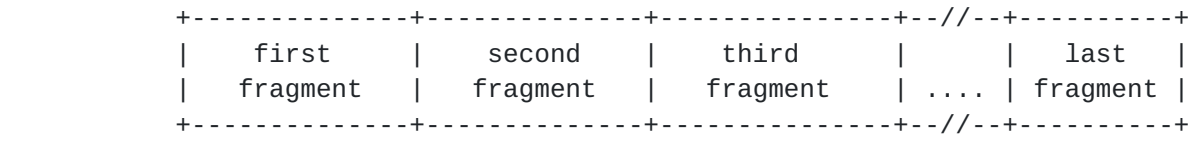
an Identification value. The Identification must be different than that of any other fragmented packet sent within the past 60 seconds (Maximum Segment Lifetime) with the same tunnel identification-- that is the same outer source and destination addresses, same UDP ports, same orig-proto, and same virtual network identifier if present.

The initial, unfragmented, and unencapsulated packet is referred to as the "original packet". This will be a layer 2 packet, layer 3 packet, or the payload of a GUE control message:

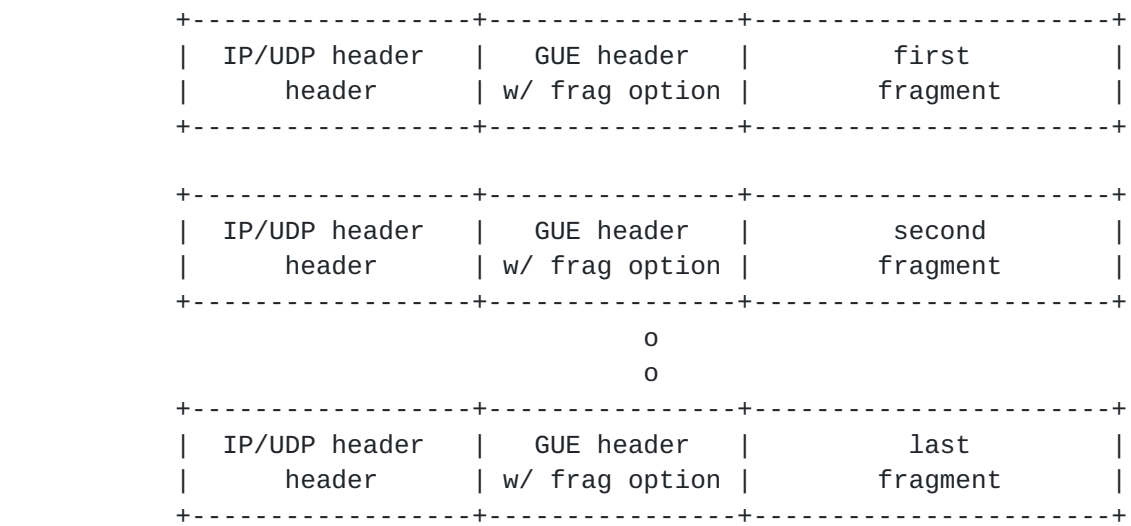


Fragmentation and encapsulation are performed on the original packet in sequence. First the packet is divided up in to fragments, and then each fragment is encapsulated. Each fragment, except possibly the last ("rightmost") one, is an integer multiple of 8 octets long. Fragments MUST be non-overlapping. The number of fragments should be minimized, and all but the last fragment should be approximately equal in length.

The fragments are transmitted in separate "fragment packets" as:



Each fragment is encapsulated as the payload of a GUE packet. This is illustrated as:



Each fragment packet is composed of:

(1) Outer IP and UDP headers as defined for GUE encapsulation.

- o The IP addresses and UDP destination port must be the same for all fragments of a fragmented packet.
- o The source port selected for the inner flow identifier must be the same value for all fragments of a fragmented packet.

(2) A GUE header that contains:

- o The C bit which is set to the same value for all the fragments of a fragmented packet based on whether a control message or data message was fragmented.
- o A proto/ctype. In the first fragment this is set to the value corresponding to the next header of the original packet and will be either an IP protocol or a control type. For subsequent fragments, this field is set to 0 for a fragmented control message or 59 (no next header) for a fragmented data messages.
- o The F bit is set and fragment option is present. See below.
- o Other GUE options. Note that options apply to the individual GUE packet. For instance, the security option would be validated before reassembly.

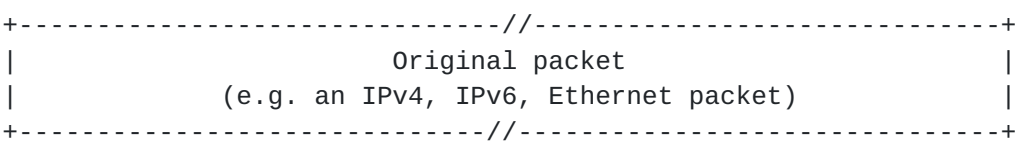
(2) The GUE fragmentation option. The option contents include:

- o Orig-proto that identifies the first header of the original packet.
- o A Fragment Offset containing the offset of the fragment, in 8-octet units, relative to the start of the of the original packet. The Fragment Offset of the first ("leftmost") fragment is 0.
- o An M flag value of 0 if the fragment is the last ("rightmost") one, else an M flag value of 1.
- o The Identification value generated for the original packet.

(3) The fragment itself.

3.2 Reassembly

At the destination, fragment packets are decapsulated and reassembled into their original, unfragmented form, as illustrated:



The following rules govern reassembly:

The IP/UDP/GUE headers of each packet are retained until all fragments have arrived. The reassembled packet is then composed of the decapsulated payloads in the GUE fragments, and the IP/UDP/GUE headers are discarded

When a GUE packet is received with the fragment option, the proto/ctype in the GUE header must be validated. In the case that the packet is a first fragment (fragment offset is zero), the proto/ctype in the GUE header must equal the orig-proto value in the fragmentation option. For subsequent fragments (fragment offset is non-zero) the proto/ctype in the GUE header must be 0 for a control message or 59 (no-next-hdr) for a data message. If the proto/ctype value is invalid then the packet MUST be dropped.

An original packet is reassembled only from GUE fragment packets that have the same outer Source Address, Destination Address, UDP source port, UDP destination port, GUE header C bit, virtual network identifier if present, orig-proto value in the fragmentation option, and Fragment Identification. The protocol type or control message type (depending on the C bit) for the reassembled packet is the value of the GUE header proto/ctype field in the first fragment.

The following error conditions may arise when reassembling fragmented packets with GUE encapsulation:

If insufficient fragments are received to complete reassembly of a packet within 60 seconds (or a configurable period) of the reception of the first-arriving fragment of that packet, reassembly of that packet must be abandoned and all the fragments that have been received for that packet must be discarded.

If the length of a fragment, as derived from the GUE fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is 1, then that fragment must be

discarded.

If the length and offset of a fragment are such that the Payload Length of the packet reassembled from that fragment would exceed 65,535 octets, then that fragment must be discarded.

If a fragment overlaps another fragment already saved for reassembly then the portion of data in the new fragment that overlaps the existing fragment must be ignored.

If the first fragment is too small then it is possible that it does not contain the necessary headers for a stateful firewall. Sending small fragments like this has been used as an attack on IP fragmentation. To mitigate this problem, an implementation should ensure that the first fragment contains the headers of the encapsulated packet at least through the transport header.

4 Security Considerations

Exploits that have been identified with IP fragmentation are conceptually applicable to GUE fragmentation.

Attacks on GUE fragmentation can be mitigated by:

- Hardened implementation that applies applicable techniques from implementation of IP fragmentation.
- Application of GUE security [I.D.hy-gue-4-secure-transport] or IPsec [RFC4301]. Security mechanisms can prevent spoofing of fragments from unauthorized sources.
- Implement fragment filter techniques for GUE encapsulation as described in [RFC1858] and [RFC3128].
- Do not accepted data in overlapping segments.
- Enforce a minimum size for the first fragment.

5 IANA Considerations

GUE fragmentation defines one flag bit in the GUE header and a corresponding 64-bit field.

6 Acknowledgements

Motivations for including an encapsulation fragment header option were discussed on the int-area mailing list in the August 2015 timeframe.

7 References

7.1 Normative References

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

[I.D.ietf-nvo3-gue] T. Herbert, L. Yong, and O. Zia, "Generic UDP Encapsulation" draft-ietf-nvo3-gue-01

7.2 Informative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

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

[RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, February 2000, <http://www.rfc-editor.org/info/rfc2764>.

[RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, October 1995, <http://www.rfc-editor.org/info/rfc1858>.

[RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, June 2001, <http://www.rfc-editor.org/info/rfc3128>.

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

[I.D.templin-aerolink] F. Templin, "Transmission of IP Packets over AERO Links" draft-templin-aerolink-62.txt

[FRAGHRM] M. Mathis, J, Heffner, and B. Chandler, "Fragmentation Considered Very Harmful", draft-mathis-frag-harmful-00

[I.D.hy-gue-4-secure-transport] L. Yong and T. Herbert, "Generic UDP Encapsulation (GUE) for Secure Transport" draft-hy-gue-4-secure-transport-02

Authors' Addresses

Tom Herbert
Facebook
Menlo Park, CA
USA

Email: tom@herbertland.com

Fred L. Templin
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org