

Network Working Group  
Internet-Draft  
Intended status: Standard Track

Lucy Yong (Ed.)  
Huawei USA  
E. Crabbe

X. Xu  
Huawei Technologies  
T. Herbert  
Facebook

Expires: September 2016

March 10, 2016

**GRE-in-UDP Encapsulation**  
**draft-ietf-tsvwg-gre-in-udp-encap-11**

**Abstract**

This document describes a method of encapsulating network protocol packets within GRE and UDP headers. The GRE-in-UDP encapsulation allows the UDP source port field to be used as an entropy field. This may be used for load balancing of GRE traffic in transit networks using existing ECMP mechanisms. This document specifies GRE-in-UDP tunnel requirements for two applicability scenarios: (1) general Internet; (2) a traffic-managed controlled environment. The controlled environment has less restrictive requirements than the general Internet.

**Status of This Document**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 10, 2016.

**Copyright Notice**

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Terminology.....</a>	<a href="#">3</a>
<a href="#">1.2.</a>	<a href="#">Requirements Language.....</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Applicability Statement.....</a>	<a href="#">4</a>
<a href="#">2.1.</a>	<a href="#">GRE-in-UDP Tunnel Usage Requirements.....</a>	<a href="#">5</a>
<a href="#">2.1.1.</a>	<a href="#">Requirements for Default GRE-in-UDP Tunnel.....</a>	<a href="#">5</a>
<a href="#">2.1.2.</a>	<a href="#">Requirements Changes for TMCE GRE-in-UDP Tunnel.....</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">GRE-in-UDP Encapsulation.....</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">IP Header.....</a>	<a href="#">9</a>
<a href="#">3.2.</a>	<a href="#">UDP Header.....</a>	<a href="#">9</a>
<a href="#">3.2.1.</a>	<a href="#">Source Port.....</a>	<a href="#">9</a>
<a href="#">3.2.2.</a>	<a href="#">Destination Port.....</a>	<a href="#">10</a>
<a href="#">3.2.3.</a>	<a href="#">Checksum.....</a>	<a href="#">10</a>
<a href="#">3.2.4.</a>	<a href="#">Length.....</a>	<a href="#">10</a>
<a href="#">3.3.</a>	<a href="#">GRE Header.....</a>	<a href="#">10</a>
<a href="#">4.</a>	<a href="#">Encapsulation Process Procedures.....</a>	<a href="#">11</a>
<a href="#">4.1.</a>	<a href="#">MTU and Fragmentation.....</a>	<a href="#">11</a>
<a href="#">4.2.</a>	<a href="#">Differentiated Services and ECN Marking.....</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Use of DTLS.....</a>	<a href="#">12</a>
<a href="#">6.</a>	<a href="#">UDP Checksum Handling.....</a>	<a href="#">12</a>
<a href="#">6.1.</a>	<a href="#">UDP Checksum with IPv4.....</a>	<a href="#">12</a>
<a href="#">6.2.</a>	<a href="#">UDP Checksum with IPv6.....</a>	<a href="#">13</a>
<a href="#">7.</a>	<a href="#">Middlebox Considerations.....</a>	<a href="#">16</a>
<a href="#">7.1.</a>	<a href="#">Middlebox Considerations for Zero Checksums.....</a>	<a href="#">17</a>
<a href="#">8.</a>	<a href="#">Congestion Considerations.....</a>	<a href="#">17</a>
<a href="#">9.</a>	<a href="#">Backward Compatibility.....</a>	<a href="#">18</a>
<a href="#">10.</a>	<a href="#">IANA Considerations.....</a>	<a href="#">19</a>
<a href="#">11.</a>	<a href="#">Security Considerations.....</a>	<a href="#">20</a>
<a href="#">12.</a>	<a href="#">Acknowledgements.....</a>	<a href="#">20</a>
<a href="#">13.</a>	<a href="#">Contributors.....</a>	<a href="#">21</a>
<a href="#">14.</a>	<a href="#">References.....</a>	<a href="#">22</a>
<a href="#">14.1.</a>	<a href="#">Normative References.....</a>	<a href="#">22</a>
<a href="#">14.2.</a>	<a href="#">Informative References.....</a>	<a href="#">23</a>
<a href="#">15.</a>	<a href="#">Authors' Addresses.....</a>	<a href="#">24</a>

## 1. Introduction

This document defines a generic GRE-in-UDP encapsulation for tunneling network protocol packets across an IP network. The encapsulation uses Generic Routing Encapsulation (GRE) [[RFC2784](#)][RFC7676] and User Datagram Protocol(UDP) [[RFC768](#)] headers. The GRE header provides payload protocol type as an EtherType in the protocol type field, and the source port field in the UDP header may be used to provide additional entropy that may be used for load balancing GRE traffic in transit networks using existing Equal-Cost Multi-Path (ECMP) mechanism. The existing ECMP mechanism is that, when the IP payload is a UDP or Transmission Control Protocol (TCP) [[RFC793](#)] packet, router hash functions frequently operate on the five-tuple of source IP address, destination IP address, UDP/TCP source port, UDP/TCP destination port, and protocol/next-header. A GRE-in-UDP tunnel offers the additional possibility of using GRE across networks that might otherwise disallow it; for instance GRE-in-UDP may be used to bridge two islands where GRE is not used natively across the Internet.

This encapsulation method requires no changes to the transit IP network. Hash functions in most existing IP routers may utilize and benefit from the use of a GRE-in-UDP tunnel without needing any change or upgrade to their ECMP implementation. The encapsulation mechanism is applicable to a variety of IP networks including Data Center and wide area networks.

GRE-in-UDP encapsulation may be used to encapsulate already tunneled traffic, i.e. tunnel-in-tunnel. In this case, GRE-in-UDP tunnel do not differentiate such end hosts from other end hosts, i.e., applying the same treatment for traffic from hosts and tunnel endpoints.

This document specifies GRE-in-UDP tunnel requirements for two applicability scenarios: (1) general Internet; (2) a traffic-managed controlled environment. The controlled environment has less restrictive requirements than the general Internet.

### 1.1. Terminology

The terms defined in [[RFC768](#)][RFC2784] are used in this document.

A traffic-managed controlled environment: an IP network that is traffic-engineered and/or otherwise managed (e.g., via use of traffic rate limiters) to avoid congestion happening.



TMCE GRE-in-UDP Tunnel: A GRE-in-UDP tunnel that can only apply to a traffic-managed controlled environment that is defined in [Section 2](#).

Default GRE-in-UDP Tunnel: A GRE-in-UDP tunnel that can apply to the general Internet.

ECMP: Equal-Cost Multi-Path

TMCE: Traffic-managed controlled environment (defined in [Section 2](#))

## 1.2. 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 [\[RFC2119\]](#).

## 2. Applicability Statement

GRE-in-UDP encapsulation applies to IPv4 and IPv6 networks. When using GRE-in-UDP encapsulation, packets encapsulated by GRE-in-UDP are treated as UDP datagrams by an IP network. As such, GRE-in-UDP tunnel needs to meet the UDP requirements specified in [\[RFC5405bis\]](#), which imposes the requirements on GRE-in-UDP tunnel usage. These requirements depend on both the delivery network and the nature of the encapsulated traffic. For example, the GRE-in-UDP tunnel protocol does not provide any congestion control functionality beyond that of the encapsulated traffic. Therefore, a GRE-in-UDP tunnel MUST be used only with congestion controlled traffic (e.g., IP traffic) and/or within a network that has traffic management capability to avoid congestion.

[\[RFC5405bis\]](#) considers two types of applicability where IETF applications utilize UDP: 1) General Internet and 2) A controlled environment. The controlled environment means a single administrative domain or bilaterally agreed connection between domains. A network forming a controlled environment can be managed/operated to meet certain conditions while the general Internet cannot be; thus the requirements for a tunnel protocol operating under a controlled environment can be less restrictive than the requirements in the general Internet.

For the purpose of this document, a traffic-managed controlled environment is defined as an IP network that is traffic-engineered and/or otherwise managed (e.g., via use of traffic rate limiters) to avoid congestion happening. The document specifies GRE-in-UDP tunnel usage in the general Internet and specifies GRE-in-UDP tunnel usage

in a traffic-managed controlled environment. Furthermore, a default GRE-in-UDP tunnel described in this document refers to the usage over the general Internet; a TMCE GRE-in-UDP tunnel described in this document refers to the usage in a traffic-managed controlled environment.

## 2.1. GRE-in-UDP Tunnel Usage Requirements

This section provides a summary of the requirements for a GRE-in-UDP tunnel. [Section 2.1.1](#) describes the default usage of GRE-in-UDP tunnel that is suitable for the general Internet; [Section 2.1.2](#) describes a set of relaxed requirements for a TMCE GRE-in-UDP tunnel used in a traffic-managed controlled environment. Both can be IPv4 or IPv6.

### 2.1.1. Requirements for Default GRE-in-UDP Tunnel

The following is a summary of the default GRE-in-UDP requirements for use over the general Internet:

1. A UDP checksum SHOULD be used when encapsulating in IPv4.
2. A UDP checksum MUST be used when encapsulating in IPv6.
3. GRE-in-UDP tunnel MUST NOT be used for traffic that does not implement congestion control. IP-traffic can be assumed to be congestion-controlled. GRE-in-UDP tunnels are not appropriate for other traffic that does not use congestion control.
4. UDP source port values that are used for flow entropy SHOULD be chosen from the ephemeral port range (49152-65535).
5. The use of the UDP source port MUST be configurable so that a single value can be set for all traffic within the tunnel (this disables use of the UDP source port to provide flow entropy). When a single value is set, a random port SHOULD be selected in order to minimize the vulnerability to off-path attacks [[RFC6056](#)].
6. For IPv6 delivery networks, the flow entropy SHOULD also be placed in the flow label field for ECMP per [[RFC6438](#)].
7. At the tunnel ingress, any fragmentation of the incoming packet (e.g., because the tunnel has an MTU that is smaller than the packet) SHOULD be performed before encapsulation [[RFC7588](#)]. In addition, the tunnel ingress MUST apply the UDP checksum to all encapsulated fragments so that the tunnel egress can validate reassembly of the fragments; it MUST set the same DSCP value to all fragments. To



avoid unwanted forwarding over multiple paths the same source UDP port value SHOULD be set in all packet fragments.

#### 2.1.2. Requirements Changes for TMCE GRE-in-UDP Tunnel

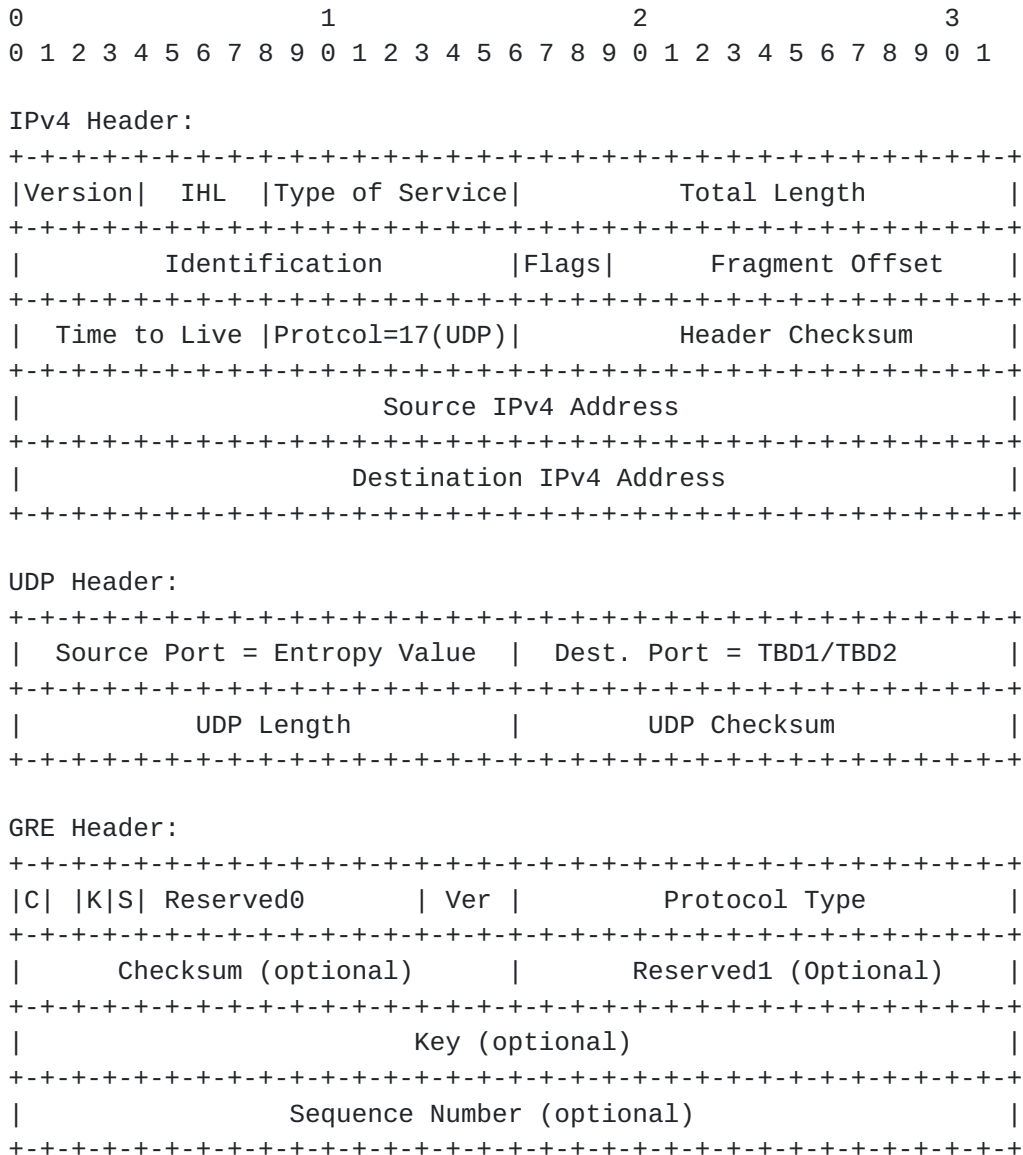
The section lists the changed requirements for a TMCE GRE-in-UDP Tunnel that applies to a traffic-managed controlled environment. This replaces requirements 1-3 listed in [Section 2.1.1](#). The requirements 4-7 in [Section 2.1.1](#) remain unchanged for a TMCE GRE-in-UDP Tunnel.

1. A UDP checksum SHOULD be used when encapsulating in IPv4. A tunnel endpoint sending GRE-in-UDP MAY disable the UDP checksum, since GRE has been designed to work without a UDP checksum [[RFC2784](#)]. However, a checksum also offers protection from mis-delivery to another port.
2. Use of UDP checksum MUST be the default when encapsulating in IPv6. This default MAY be overridden via configuration of UDP zero-checksum mode. All usage of UDP zero-checksum mode with IPv6 is subject to the additional requirements specified in [Section 6.2](#).
3. A GRE-in-UDP tunnel MAY encapsulate traffic that is not congestion controlled.

### 3. GRE-in-UDP Encapsulation

The GRE-in-UDP encapsulation format contains UDP header [[RFC768](#)] and GRE header [[RFC2890](#)]. The format is shown as follows: (presented in bit order)

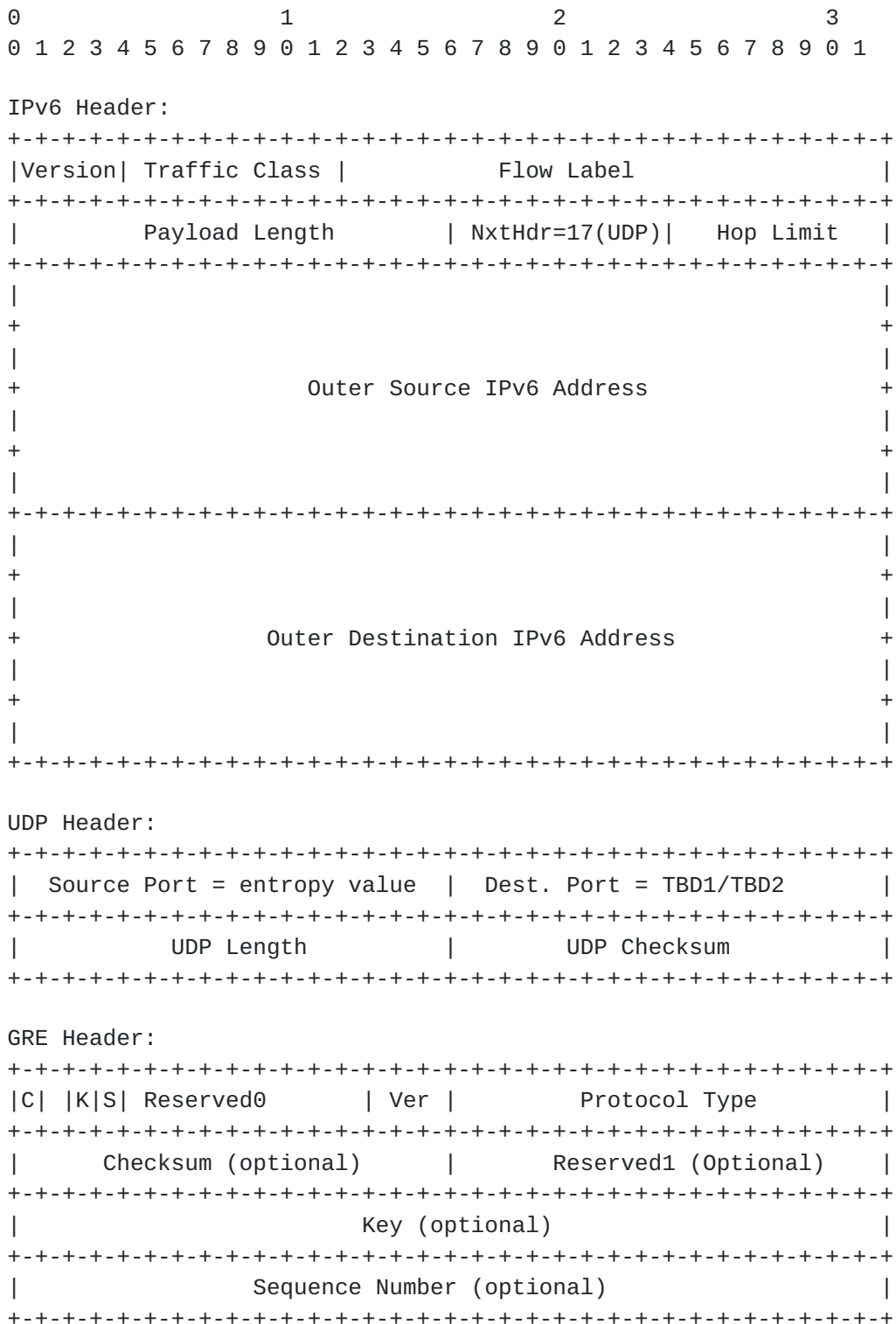




IANA Note: Replace TBD1 and TBD2 with the IANA-assigned numbers

Figure 1 UDP+GRE Headers in IPv4





IANA Note: Replace TBD1 and TBD2 with the IANA-assigned numbers

Figure 2 UDP+GRE Headers in IPv6



The contents of the IP, UDP, and GRE headers that are relevant in this encapsulation are described below.

### 3.1. IP Header

An encapsulator **MUST** encode its own IP address as the source IP address and the decapsulator's IP address as the destination IP address. A sufficiently large value is needed in the IPv4 TTL field or IPv6 Hop Count field to allow delivery of the encapsulated packet to the peer of the encapsulation.

### 3.2. UDP Header

#### 3.2.1. Source Port

GRE-in-UDP permits the UDP source port value to be used to encode an entropy value. The UDP source port contains a 16-bit entropy value that is generated by the encapsulator to identify a flow for the encapsulated packet. The port value **SHOULD** be within the ephemeral port range, i.e., 49152 to 65535, where the high order two bits of the port are set to one. This provides fourteen bits of entropy for the inner flow identifier. In the case that an encapsulator is unable to derive flow entropy from the payload header or the entropy usage has to be disabled to meet operational requirements (see [Section 7](#)), to avoid reordering with a packet flow, the encapsulator **SHOULD** use the same UDP source port value for all packets assigned to a flow e.g., the result of an algorithm that perform a hash of the tunnel ingress and egress IP address.

The source port value for a flow set by an encapsulator **MAY** change over the lifetime of the encapsulated flow. For instance, an encapsulator may change the assignment for Denial of Service (DOS) mitigation or as a means to effect routing through the ECMP network. An encapsulator **SHOULD NOT** change the source port selected for a flow more than once every thirty seconds.

Note: An IPv6 tunnel endpoint should copy a flow entropy value in the IPv6 flow label field (requirement 6). This permits network equipment to inspect this value and utilize it during forwarding, e.g. to perform ECMP [[RFC6438](#)].

This document places requirements on the generation of the flow entropy value but does not specify the algorithm that an implementation should use to derive this value.



### 3.2.2. Destination Port

The destination port of the UDP header is set either GRE-in-UDP (TBD1) or GRE-UDP-DTLS (TBD2) (see [Section 5](#)). IANA Note: Please replace TBD1 and TBD2 with the IANA-assigned numbers.

### 3.2.3. Checksum

The UDP checksum is set and processed per [\[RFC768\]](#) and [\[RFC1122\]](#) for IPv4, and [\[RFC2460\]](#) for IPv6. Requirements for checksum handling and use of zero UDP checksums are detailed in [Section 6](#).

### 3.2.4. Length

The usage of this field is in accordance with the current UDP specification in [\[RFC768\]](#). This length will include the UDP header (eight bytes), GRE header, and the GRE payload (encapsulated packet).

## 3.3. GRE Header

An encapsulator sets the protocol type (EtherType) of the packet being encapsulated in the GRE Protocol Type field.

An encapsulator MAY set the GRE Key Present, Sequence Number Present, and Checksum Present bits and associated fields in the GRE header as defined by [\[RFC2784\]](#) and [\[RFC2890\]](#). Usage of the reserved bits, i.e., Reserved0, is specified in [\[RFC2784\]](#).

The GRE checksum MAY be enabled to protect the GRE header and payload. When the UDP checksum is enabled, it protects the GRE payload, resulting in the GRE checksum being mostly redundant. Enabling both checksums may result in unnecessary processing. Since the UDP checksum covers the pseudo-header and the packet payload, including the GRE header and its payload, the UDP checksum SHOULD be used in preference to using the GRE checksum.

An implementation MAY use the GRE keyid to authenticate the encapsulator. (See Security Section) In this model, a shared value is either configured or negotiated between an encapsulator and decapsulator. When a decapsulator determines a presented keyid is not valid for the source, the packet MUST be dropped.

Although GRE-in-UDP encapsulation protocol uses both UDP header and GRE header, it is one tunnel encapsulation protocol. GRE and UDP headers MUST be applied and removed as a pair at the encapsulation and decapsulation points. This specification does not support UDP encapsulation of a GRE header where that GRE header is applied or





removed at a network node other than the UDP tunnel ingress or egress.

#### **4. Encapsulation Process Procedures**

The procedures specified in this section apply to both a default GRE-in-UDP tunnel and a TMCE GRE-in-UDP tunnel.

The GRE-in-UDP encapsulation allows encapsulated packets to be forwarded through "GRE-in-UDP tunnels". The encapsulator **MUST** set the UDP and GRE header according to [Section 3](#).

Intermediate routers, upon receiving these UDP encapsulated packets, could load balance these packets based on the hash of the five-tuple of UDP packets.

Upon receiving these UDP encapsulated packets, the decapsulator decapsulates them by removing the UDP and GRE headers and then processes them accordingly.

GRE-in-UDP allows encapsulation of unicast, IPv4 broadcast, or multicast traffic. Entropy may be generated from the header of encapsulated packets at an encapsulator. The mapping mechanism between the encapsulated multicast traffic and the multicast capability in the IP network is transparent and independent to the encapsulation and is otherwise outside the scope of this document.

To provide entropy for ECMP, GRE-in-UDP does not rely on GRE keep-alive. It is RECOMMENDED not to use GRE keep-alive in the GRE-in-UDP tunnel. This aligns with middlebox traversal guidelines in [Section 3.5](#) of [\[RFC5405bis\]](#).

##### **4.1. MTU and Fragmentation**

Regarding packet fragmentation, an encapsulator/decapsulator **SHOULD** be compliant with [\[RFC7588\]](#) and perform fragmentation before the encapsulation. The size of fragments **SHOULD** be less or equal to the PMTU associated with the path between the GRE ingress and the GRE egress tunnel endpoints minus the GRE and UDP overhead, assuming the egress resemble MTU is larger than PMTU. When applying payload fragmentation, the UDP checksum **MUST** be used so that the receiving endpoint can validate reassembly of the fragments; the same src UDP port **SHOULD** be used for all packet fragments to ensure the transit routers will forward the fragments on the same path.

If the operator of the transit network supporting the tunnel is able to control the payload MTU size, the MTU **SHOULD** be configured to



avoid fragmentation, i.e., sufficient for the largest supported size of packet, including all additional bytes introduced by the tunnel overhead [[RFC5405bis](#)].

#### 4.2. Differentiated Services and ECN Marking

To ensure that tunneled traffic receives the same treatment over the IP network, prior to the encapsulation process, an encapsulator processes the tunneled IP packet headers to retrieve appropriate parameters for the encapsulating IP packet header such as DiffServ [[RFC2983](#)]. Encapsulation end points that support Explicit Congestion Notification (ECN) must use the method described in [[RFC6040](#)] for ECN marking propagation. The congestion control process is outside of the scope of this document.

Additional information on IP header processing is provided in [Section 3.1](#).

### 5. Use of DTLS

Datagram Transport Layer Security (DTLS) [[RFC6347](#)] can be used for application security and can preserve network and transport layer protocol information. Specifically, if DTLS is used to secure the GRE-in-UDP tunnel, the destination port of the UDP header MUST be set to an IANA-assigned value (TBD2) indicating GRE-in-UDP with DTLS, and that UDP port MUST NOT be used for other traffic. The UDP source port field can still be used to add entropy, e.g., for load-sharing purposes. DTLS usage is limited to a single DTLS session for any specific tunnel encapsulator/ decapsulator pair (identified by source and destination IP addresses). Both IP addresses MUST be unicast addresses - multicast traffic is not supported when DTLS is used. A GRE-in-UDP tunnel decapsulator that supports DTLS is expected to be able to establish DTLS sessions with multiple tunnel encapsulators, and likewise an GRE-in-UDP tunnel encapsulator is expected to be able to establish DTLS sessions with multiple decapsulators (although different source and/or destination IP addresses may be involved (see [Section 6.2](#)) for discussion of one situation where use of different source IP addresses is important).

IANA Note: Please replace TBD2 with the IANA-assigned numbers.

### 6. UDP Checksum Handling

#### 6.1. UDP Checksum with IPv4

For UDP in IPv4, the UDP checksum MUST be processed as specified in [[RFC768](#)] and [[RFC1122](#)] for both transmit and receive. The IPv4



header includes a checksum which protects against mis-delivery of the packet due to corruption of IP addresses. The UDP checksum potentially provides protection against corruption of the UDP header, GRE header, and GRE payload. Disabling the use of checksums is a deployment consideration that should take into account the risk and effects of packet corruption.

When a decapsulator receives a packet, the UDP checksum field **MUST** be processed. If the UDP checksum is non-zero, the decapsulator **MUST** verify the checksum before accepting the packet. By default a decapsulator **SHOULD** accept UDP packets with a zero checksum. A node **MAY** be configured to disallow zero checksums per [\[RFC1122\]](#); this may be done selectively, for instance disallowing zero checksums from certain hosts that are known to be sending over paths subject to packet corruption. If verification of a non-zero checksum fails, a decapsulator lacks the capability to verify a non-zero checksum, or a packet with a zero-checksum was received and the decapsulator is configured to disallow, the packet **MUST** be dropped and an event **MAY** be logged.

## 6.2. UDP Checksum with IPv6

For UDP in IPv6, the UDP checksum **MUST** be processed as specified in [\[RFC768\]](#) and [\[RFC2460\]](#) for both transmit and receive.

When UDP is used over IPv6, the UDP checksum is relied upon to protect both the IPv6 and UDP headers from corruption. As such, A default GRE-in-UDP Tunnel **MUST** perform UDP checksum; A TMCE GRE-in-UDP Tunnel **MAY** be configured with the UDP zero-checksum mode if the traffic-managed controlled environment or a set of closely cooperating traffic-managed controlled environments (such as by network operators who have agreed to work together in order to jointly provide specific services) meet at least one of following conditions:

- a. It is known (perhaps through knowledge of equipment types and lower layer checks) that packet corruption is exceptionally unlikely and where the operator is willing to take the risk of undetected packet corruption.
- b. It is judged through observational measurements (perhaps of historic or current traffic flows that use a non-zero checksum) that the level of packet corruption is tolerably low and where the operator is willing to take the risk of undetected packet corruption.



- c. Carrying applications that are tolerant of mis-delivered or corrupted packets (perhaps through higher layer checksum, validation, and retransmission or transmission redundancy) where the operator is willing to rely on the applications using the tunnel to survive any corrupt packets.

The following requirements apply to a TMCE GRE-in-UDP tunnel that use UDP zero-checksum mode:

- a. Use of the UDP checksum with IPv6 MUST be the default configuration of all GRE-in-UDP tunnels.
- b. The GRE-in-UDP tunnel implementation MUST comply with all requirements specified in [Section 4 of \[RFC6936\]](#) and with requirement 1 specified in [Section 5 of \[RFC6936\]](#).
- c. The tunnel decapsulator SHOULD only allow the use of UDP zero-checksum mode for IPv6 on a single received UDP Destination Port regardless of the encapsulator. The motivation for this requirement is possible corruption of the UDP Destination Port, which may cause packet delivery to the wrong UDP port. If that other UDP port requires the UDP checksum, the mis-delivered packet will be discarded.
- d. It is RECOMMENDED that the UDP zero-checksum mode for IPv6 is only enabled for certain selected source addresses. The tunnel decapsulator MUST check that the source and destination IPv6 addresses are valid for the GRE-in-UDP tunnel on which the packet was received if that tunnel uses UDP zero-checksum mode and discard any packet for which this check fails.
- e. The tunnel encapsulator SHOULD use different IPv6 addresses for each GRE-in-UDP tunnel that uses UDP zero-checksum mode regardless of the decapsulator in order to strengthen the decapsulator's check of the IPv6 source address (i.e., the same IPv6 source address SHOULD NOT be used with more than one IPv6 destination address, independent of whether that destination address is a unicast or multicast address). When this is not possible, it is RECOMMENDED to use each source IPv6 address for as few UDP zero-checksum mode GRE-in-UDP tunnels as is feasible.
- f. When any middlebox exists on the path of a GRE-in-UDP tunnel, it is RECOMMENDED to use the default mode, i.e. use UDP checksum, to reduce the chance that the encapsulated packets to be dropped.





- g. Any middlebox that allows the UDP zero-checksum mode for IPv6 MUST comply with requirement 1 and 8-10 in [Section 5 of \[RFC6936\]](#).
- h. Measures SHOULD be taken to prevent IPv6 traffic with zero UDP checksums from "escaping" to the general Internet; see [Section 8](#) for examples of such measures.
- i. IPv6 traffic with zero UDP checksums MUST be actively monitored for errors by the network operator. For example, the operator may monitor Ethernet layer packet error rates.
- j. If a packet with a non-zero checksum is received, the checksum MUST be verified before accepting the packet. This is regardless of whether the tunnel encapsulator and decapsulator have been configured with UDP zero-checksum mode.

The above requirements do not change either the requirements specified in [\[RFC2460\]](#) as modified by [\[RFC6935\]](#) or the requirements specified in [\[RFC6936\]](#).

The requirement to check the source IPv6 address in addition to the destination IPv6 address, plus the strong recommendation against reuse of source IPv6 addresses among GRE-in-UDP tunnels collectively provide some mitigation for the absence of UDP checksum coverage of the IPv6 header. A traffic-managed controlled environment that satisfies at least one of three conditions listed above in this section provides additional assurance.

A GRE-in-UDP tunnel is suitable for transmission over lower layers in the traffic-managed controlled environments that are allowed by the exceptions stated above and the rate of corruption of the inner IP packet on such networks is not expected to increase by comparison to GRE traffic that is not encapsulated in UDP. For these reasons, GRE-in-UDP does not provide an additional integrity check except when GRE checksum is used when UDP zero-checksum mode is used with IPv6, and this design is in accordance with requirements 2, 3 and 5 specified in [Section 5 of \[RFC6936\]](#).

Generic Router Encapsulation (GRE) does not accumulate incorrect state as a consequence of GRE header corruption. A corrupt GRE packet may result in either packet discard or forwarding of the packet without accumulation of GRE state. Active monitoring of GRE-in-UDP traffic for errors is REQUIRED as occurrence of errors will result in some accumulation of error information outside the protocol for operational and management purposes. This design is in accordance with requirement 4 specified in [Section 5 of \[RFC6936\]](#).



The remaining requirements specified in [Section 5 of \[RFC6936\]](#) are not applicable to GRE-in-UDP. Requirements 6 and 7 do not apply because GRE does not include a control feedback mechanism. Requirements 8-10 are middlebox requirements that do not apply to GRE-in-UDP tunnel endpoints (see [Section 7.1](#) for further middlebox discussion).

It is worth mentioning that the use of a zero UDP checksum should present the equivalent risk of undetected packet corruption when sending similar packet using GRE-in-IPv6 without UDP [\[RFC7676\]](#) and without GRE checksums.

In summary, a TMCE GRE-in-UDP Tunnel is allowed to use UDP-zero-checksum mode for IPv6 when the conditions and requirements stated above are met. Otherwise the UDP checksum need to be used for IPv6 as specified in [\[RFC768\]](#) and [\[RFC2460\]](#). Use of GRE checksum is RECOMMENDED when the UDP checksum is not used.

## **7. Middlebox Considerations**

Many middleboxes read or update UDP port information of the packets that they forward. Network Address/Port Translator (NAPT) is the most commonly deployed Network Address Translation (NAT) device [\[RFC4787\]](#). An NAPT device establishes a NAT session to translate the {private IP address, private source port number} tuple to a {public IP address, public source port number} tuple, and vice versa, for the duration of the UDP session. This provides a UDP application with the "NAT-pass-through" function. NAPT allows multiple internal hosts to share a single public IP address. The port number, i.e., the UDP Source Port number, is used as the demultiplexer of the multiple internal hosts. However, the above NAPT behaviors conflict with the behavior a GRE-in-UDP tunnel that is configured to use the UDP source port value to provide entropy.

A GRE-in-UDP tunnel is unidirectional; the tunnel traffic is not expected to be returned back to the UDP source port values used to generate entropy. However some middleboxes (e.g., firewall) assume that bidirectional traffic uses a common pair of UDP ports. This assumption also conflicts with the use of the UDP source port field as entropy.

Hence, use of the UDP source port for entropy may impact middleboxes behavior. If a GRE-in-UDP tunnel is expected to be used on a path with a middlebox, the tunnel can be configured to either disable use of the UDP source port for entropy or to configure middleboxes to pass packets with UDP source port entropy.



### 7.1. Middlebox Considerations for Zero Checksums

IPv6 datagrams with a zero UDP checksum will not be passed by any middlebox that validates the checksum based on [\[RFC2460\]](#) or that updates the UDP checksum field, such as NATs or firewalls. Changing this behavior would require such middleboxes to be updated to correctly handle datagrams with zero UDP checksums. The GRE-in-UDP encapsulation does not provide a mechanism to safely fall back to using a checksum when a path change occurs redirecting a tunnel over a path that includes a middlebox that discards IPv6 datagrams with a zero UDP checksum. In this case the GRE-in-UDP tunnel will be black-holed by that middlebox.

As such, when any middlebox exists on the path of GRE-in-UDP tunnel, it is RECOMMENDED to use the UDP checksum to increase the probability of successful transmission of GRE-in-UDP packets. Recommended changes to allow firewalls, NATs and other middleboxes to support use of an IPv6 zero UDP checksum are described in [Section 5 of \[RFC6936\]](#).

## 8. Congestion Considerations

Section 3.1.9 of [\[RFC5405bis\]](#) discussed the congestion implications of UDP tunnels. As discussed in [\[RFC5405bis\]](#), because other flows can share the path with one or more UDP tunnels, congestion control [\[RFC2914\]](#) needs to be considered.

The impact of congestion must be considered both in terms of the effect on the rest of the network containing a UDP, and in terms of the effect on the flows using the UDP tunnels. The potential impact of congestion from a UDP tunnel depends upon what sort of traffic is carried over the tunnel, as well as the path of the tunnel.

In many cases, a GRE-in-UDP tunnel is used to carry IP traffic. IP traffic is generally assumed to be congestion controlled, and thus a tunnel carrying general IP traffic generally does not need additional congestion control mechanisms.

A default GRE-in-UDP tunnel can be used to carry IP traffic that is known to be congestion controlled on the Internet. Internet IP traffic is generally assumed to be congestion-controlled. The default usage MUST NOT be used over the general Internet, or over non-cooperating network operators, to carry traffic that is not congestion-controlled.

A TMCE GRE-in-UDP tunnel can be used to carry traffic that is not necessarily congestion controlled. For example, GRE-in-UDP may be



used to carry MPLS that carries pseudowire or VPN traffic where specific bandwidth guarantees are provided to each pseudowire or to each VPN. In such cases, network operators may avoid congestion by careful provisioning of their networks, by rate limiting of user data traffic, and traffic engineering according to path capacity. For this reason, when a TMCE GRE-in-UDP tunnel carries this type of traffic, the usage MUST be constrained to a traffic-managed controlled environment (e.g., single operator network that utilizes careful provisioning (e.g., rate limiting at the entries of the network while over-provisioning network capacity) to manage congestion, or within a limited number of networks whose operators closely cooperate in order to jointly provide this same careful provisioning.

When a TMCE GRE-in-UDP tunnel is used to carry the traffic that is not necessary congestion controlled, measures SHOULD be taken to prevent non-congestion-controlled GRE-in-UDP traffic from "escaping" to the general Internet, e.g.:

- o Physical or logical isolation of the links carrying GRE-in-UDP from the general Internet.
- o Deployment of packet filters that block the UDP ports assigned for GRE-in-UDP.
- o Imposition of restrictions on GRE-in-UDP traffic by software tools used to set up GRE-in-UDP tunnels between specific end systems (as might be used within a single data center). For examples, a GRE-in-UDP tunnel only carries IP traffic or a GRE-in-UDP tunnel supports NVGRE encapsulation [[RFC7637](#)] only (Although the payload type is Ethernet in NVGRE, NVGRE protocol mandates that the payload of Ethernet is IP).
- o Use of a "Circuit Breaker" for the tunneled traffic as described in [[CB](#)].

## **9. Backward Compatibility**

In general, tunnel ingress routers have to be upgraded in order to support the encapsulations described in this document.

No change is required at transit routers to support forwarding of the encapsulation described in this document.

If a tunnel endpoint (a host or router) that is intended for use as a decapsulator does not support or enable the GRE-in-UDP

encapsulation described in this document, it is not that an endpoint will listen on the destination port assigned to the GRE-encapsulation (TBD1 and TBD2). In these cases, the endpoint will perform normal UDP processing and respond to an encapsulator with an ICMP message indicating "port unreachable" according to [\[RFC792\]](#). Upon receiving this ICMP message, the node MUST NOT continue to use GRE-in-UDP encapsulation toward this peer without management intervention.

IANA NOTE: Please replace TBD1 and TBD2 with the IANA-assigned numbers.

## **10. IANA Considerations**

IANA is requested to make the following allocations:

One UDP destination port number for the indication of GRE

Service Name: GRE-in-UDP  
Transport Protocol(s): UDP  
Assignee: IESG <iesg@ietf.org>  
Contact: IETF Chair <chair@ietf.org>  
Description: GRE-in-UDP Encapsulation  
Reference: [This.I-D]  
Port Number: TBD1  
Service Code: N/A  
Known Unauthorized Uses: N/A  
Assignment Notes: N/A

One UDP destination port number for the indication of GRE with DTLS

Service Name: GRE-UDP-DTLS  
Transport Protocol(s): UDP  
Assignee: IESG <iesg@ietf.org>  
Contact: IETF Chair <chair@ietf.org>  
Description: GRE-in-UDP Encapsulation with DTLS  
Reference: [This.I-D]  
Port Number: TBD2  
Service Code: N/A  
Known Unauthorized Uses: N/A  
Assignment Notes: N/A



## **11. Security Considerations**

GRE-in-UDP encapsulation does not affect security for the payload protocol. When using GRE-in-UDP, Network Security in a network is mostly equivalent to that of a network using GRE.

To secure original traffic, DTLS SHOULD be used. (See [Section 5](#))

In the case that UDP source port for entropy usage is disabled, a random port SHOULD be selected in order to minimize the vulnerability to off-path attacks.[\[RFC6056\]](#) The random port may also be periodically changed to mitigate certain denial of service attacks as mentioned in [Section 3.2.1](#).

Using one standardized value as the UDP destination port for an encapsulation indication may increase the vulnerability of off-path attack. To overcome this, an alternate port may be agreed upon to use between an encapsulator and decapsulator [\[RFC6056\]](#). How the encapsulator end points communicate the value is outside scope of this document.

This document does not require that a decapsulator validates the IP source address of the tunneled packets (with the exception that the IPv6 source address MUST be validated when UDP zero-checksum mode is used with IPv6), but it should be understood that failure to do so presupposes that there is effective destination-based (or a combination of source-based and destination-based) filtering at the boundaries.

Corruption of a GRE header can cause a privacy and security concern for some applications that rely on the key field for traffic segregation. When GRE key field is used for privacy and security, either UDP checksum or GRE checksum SHOULD be used for GRE-in-UDP with both IPv4 and IPv6, and in particular, when UDP zero-checksum mode is used, GRE checksum SHOULD be used.

## **12. Acknowledgements**

Authors like to thank Vivek Kumar, Ron Bonica, Joe Touch, Ruediger Geib, Lar Edds, Lloyd Wood, Bob Briscoe, and many others for their review and valuable input on this draft.

Thank the design team led by David Black (members: Ross Callon, Gorry Fairhurst, Xiaohu Xu, Lucy Yong) to efficiently work out the

descriptions for the congestion considerations and IPv6 UDP zero checksum.

Thank David Black and Gorrry Fairhurst for their great help in document editing.

### **13. Contributors**

The following people all contributed significantly to this document and are listed below in alphabetical order:

David Black  
EMC Corporation  
176 South Street  
Hopkinton, MA 01748  
USA

Email: david.black@emc.com

Ross Callon  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
USA

Email: rcallon@juniper.net

John E. Drake  
Juniper Networks

Email: jdrake@juniper.net

Gorrry Fairhurst  
University of Aberdeen

Email: gorrry@erg.abdn.ac.uk

Yongbing Fan  
China Telecom  
Guangzhou, China.  
Phone: +86 20 38639121

Email: fanyb@gsta.com

Adrian Farrel  
Juniper Networks

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

Vishwas Manral  
Hewlett-Packard Corp.  
3000 Hanover St, Palo Alto.

Email: [vishwas.manral@hp.com](mailto:vishwas.manral@hp.com)

Carlos Pignataro  
Cisco Systems  
7200-12 Kit Creek Road  
Research Triangle Park, NC 27709 USA

EMail: [cpignata@cisco.com](mailto:cpignata@cisco.com)

## **14. References**

### 14.1. Normative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [RFC1122] Braden, R., "Requirements for Internet Hosts -- Communication Layers", [RFC1122](#), October 1989.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC2119](#), March 1997.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", [RFC 2784](#), March 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", [RFC2890](#), September 2000.
- [RFC5405bis] Eggert, L., "Unicast UDP Usage Guideline for Application Designers", [draft-ietf-tsvwg-rfc5405bis](#), work in progress.

- [RFC6040] Briscoe, B., "Tunneling of Explicit Congestion Notification", [RFC6040](#), November 2010.
- [RFC6347] Rescoria, E., Modadugu, N., "Datagram Transport Layer Security Version 1.2", [RFC6347](#), 2012.
- [RFC6438] Carpenter, B., Amante, S., "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in tunnels", [RFC6438](#), November, 2011.
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", [RFC 6935](#), April 2013.
- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", [RFC 6936](#), April 2013.

#### 14.2. Informative References

- [RFC792] Postel, J., "Internet Control Message Protocol", STD 5, [RFC 792](#), September 1981.
- [RFC793] DARPA, "Transmission Control Protocol", [RFC793](#), September 1981.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2914] Floyd, S., "Congestion Control Principles", [RFC2914](#), September 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", [RFC2983](#), October 2000.
- [RFC4787] Audet, F., et al, "network Address Translation (NAT) Behavioral Requirements for Unicast UDP", [RFC4787](#), January 2007.
- [RFC6056] Larsen, M. and Gont, F., "Recommendations for Transport-Protocol Port Randomization", [RFC6056](#), January 2011.
- [RFC6438] Carpenter, B., Amante, S., "Using the Ipv6 Flow Label for Equal Cost Multipath Routing and Link Aggreation in Tunnels", [RFC6438](#), November 2011.
- [RFC7588] Bonica, R., "A Fragmentation Strategy for Generic Routing Encapsulation (GRE)", [RFC7588](#), July 2015.



- [RFC7637] Garg, P. and Wang, Y., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", [RFC7637](#), September 2015.
- [RFC7676] Pignataro, C., Bonica, R., Krishnan, S., "IPv6 Support for Generic Routing Encapsulation (GRE)", [RFC7676](#), October 2015.
- [CB] Fairhurst, G., "Network Transport Circuit Breakers", [draft-ietf-tsvwg-circuit-breaker-13](#), work in progress.

## **15. Authors' Addresses**

Edward Crabbe

Email: [edward.crabbe@gmail.com](mailto:edward.crabbe@gmail.com)

Lucy Yong  
Huawei Technologies, USA

Email: [lucy.yong@huawei.com](mailto:lucy.yong@huawei.com)

Xiaohu Xu  
Huawei Technologies,  
Beijing, China

Email: [xuxiaohu@huawei.com](mailto:xuxiaohu@huawei.com)

Tom Herbert  
Facebook  
1 Hacker Way  
Menlo Park, CA  
Email : [tom@herbertland.com](mailto:tom@herbertland.com)