Internet-Draft
Intended status: Standard track
Expires April 4, 2019

T. Herbert
Quantonium

October 1, 2018

Control Messages for Generic UDP Encapsulation
draft-herbert-intarea-gue-ctrl-messages-00

Status of this Memo

Copyright Notice

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

Abstract

This specification defines a set of basic control messages for Generic UDP Encapsulation (GUE). One pair of messages provides a means to query the GUE capabilities of a peer, another pair defines an echo request and response exchange for testing reachability.

Table of Contents

1 Introduction

This specification describes some basic control messages for Generic UDP Encapsulation (GUE). A capabilities query message and response message are defined for a node to query a peer GUE node for supported capabilities. Echo request and echo reply control messages are defined to verify reachability and measure latency to a GUE peer node.

The capabilities query is used to ascertain the capabilities of a peer for receiving GUE messages. For instance, a node may query a GUE peer to determine what GUE variants it supports, or what flags are supported for GUE variant 0. A capabilities query control message and a capabilities response control message are defined. A response message indicates the capabilities for receiving GUE messages. A node may send a capabilities query to a peer GUE node and based on the response it may subsequently use supported flags, optional extensions, GUE variants, or control messages when sending GUE messages to the peer.

Echo request and response messages are used to test for reachability and liveness of a GUE peer node. A node sends an echo request control message and a peer will respond with an echo reply control message. Upon receiving an echo reply, reachability to the GUE peer node is considered verified. The echo request includes arbitrary data that is reflected by the peer in an echo reply. The echo data may contain a timestamp and identifier to perform round trip latency measurement.

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 [RFC2119].

2 GUE capabilities query and response messages

This section describes the GUE capabilities query and response messages.

2.1 Capabilities query message

A GUE capabilities query message has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
   | 0 |1|   Hlen  |      0x1      |             Flags             | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ G
   |                                                               | U
   ~                  Extensions Fields (optional)                 ~ E
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ h
   |                                                               | d
   ~                    Private data (optional)                    ~ r
   |                                                               |/
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                     Query identifier                          +
   |                                                               |
   +                                                               +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Pertinent GUE header fields are:

- o C bit: Set to 1 to indicate a control message

- o Proto/ctype: Set to 0x1 to indicate a capabilities query message

GUE flags, extension fields, and private data SHOULD NOT be used in a capabilities query message.

Control message fields are:

- o Query identifier: Used to match queries with responses. This is set to a different non-zero random value in each query.

2.2 Capabilities response message

A GUE capabilities response message has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
   | 0 |1|   Hlen  |      0x2      |             Flags             | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ G
   |                                                               | U
   ~                  Extensions Fields (optional)                 ~ E
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ h
   |                                                               | d
   ~                    Private data (optional)                    ~ r
   |                                                               |/
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                     Query identifier                          +
   |                                                               |
   +                                                               +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                    Capabilities TLVs                          ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Pertinent GUE header fields are:

- o C bit: Set to 1 to indicate a control message

- o Proto/ctype: Set to 0x2 to indicate a capabilities response message

GUE flags, extension fields, and private data SHOULD NOT be used in a capabilities response message.

Control message fields are:

- o Query identifier: Reflected value from the capabilities query message

- o Capabilities TLVs: A set of Type Length Value (TLV) structures that describe the capabilities of the reporting node

2.3 Capabilities TLV format

Capabilities TLVs have the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                          Value                                ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fields:

- o Type: Type for TLV. Defined types are described below

- o Length: Length in bytes of a TLV Value. Note that this length does not include the two bytes for Type and Length.

- o Value: Data for the TLV

2.4 TLV types

The table below lists the TLVs defined in this document. The "Length" column indicates any required limits on TLVs, and the "Typical Length" column indicates the most useful lengths for the TLV.

Type	Length	Typical Length	Meaning
0			RESERVED
1	variable	1	GUE variants
2	variable	1 to 32	Control message types
3	variable	2 (currently)	GUE flags/extensions
4	variable	1 to 32	Payload transform types
5-126			UNASSIGNED (assignable by IANA)
127-255			User defined

2.4.1 GUE variants

This TLV reports the GUE variants that are support by a node.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0x1    |  Length (1)   |    Bit map         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TLV data is a variable length bit map of supported GUE variants. Bit 0 in the data indicates variant 0 is supported, bit 1 in the data indicates variant 1 is supported, etc. GUE allows up to four variants where variants 0 and 1 have been defined, so only the first four bits in the map are meaningful. If bits are set the map after the fourth bit they are ignored. Similarly, any bytes in the data beyond the first byte are ignored.

Variant 0 must be supported so bit 0 should always be set.

2.4.2 Control message types

This TLV reports the control message types that are supported by a node.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0x2    | Length (1-32) |    Bit map          ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TLV data is a bit map of supported control message types. Bit 0 in the data indicates control message 0 is supported, bit 1 in the data indicates control message 1 is supported, etc. The range of values for a control message type is 0 to 255, so the maximum useful length of the TLV data is sixteen bytes. If the data length is greater than sixteen bytes then the additional bytes are ignored.

The bit for control message 0 should be set since that value is used to indicate that the payload cannot be parsed as a control message [GUE]. Control message 1 should also be marked as supported given that fact the capabilities represented in the TLV are sent in response to a capabilities query control message which has type of 1.

2.4.3 GUE flags

This TLV reports the GUE flags that are supported by a node. In the case that flags refer to option extensions, the TLV indicates support for the extensions.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0x3    |  Length (>=2) |        Bit map   ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TLV data is a bit map of supported GUE flags. Bit 0 in the data indicates the flag corresponding to bit 0 of GUE flags is supported, bit 1 indicates the flag for the second bit in GUE flags is supported, etc.

For a multi-bit (paired) flag, the corresponding bits in the bit map are taken to be the maximum value supported for the multi-bit flag. For instance, with a three bit flag, a value of 0x2 indicates flag combinations 0x1 and 0x2 are supported. A value of 0x7 indicates that all seven combinations are supported. If more granularity is needed, for example only values 0x1 and 0x3 are supported, then an additional TLV can be defined to described supported combinations of a multi-bit flag.

2.4.4 Payload transform types

This TLV reports the types of the Payload Transform optional extension that a node supports.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0x3    | Length (1-32) |        Bit map   ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TLV data is a bit map of supported payload transform types. Bit 0 in the data indicates payload transform type 0 is supported, bit 1 in the data indicates payload transform type 1 is supported, etc. The range of values for a payload transform type is 0 to 255, so the maximum useful length of the TLV data is sixteen bytes. If the data length is greater than sixteen bytes then the additional bytes are ignored.

2.5 Operation

This section describes the operation of capabilities query and response messages.

2.5.1 Sending a capabilities query

A GUE node MAY send a capabilities request to a peer. The request is a well formatted GUE control message. The Query Identifier MUST be set to a non-zero value and SHOULD random. The sender SHOULD save the Query Identifier in a query context to match a response. The sender SHOULD set a timer to receive a response. If no response is received before timeout, then the request context is released.

2.5.2 Receiving a capabilities query

When a node receives a capabilities query it MAY send a response message. The Query Identifier that was received in the query message is reflected in response. The responding node creates the TLVs for capabilities that it wishes to report. A node is not obligated to report all implemented capabilities and may tailor its response per the identity of the requestor. It may withhold reporting of capabilities for security reasons; for instance, the security option is only useful between two peers if keys are negotiated out of band so indicating support in a capabilities response is not necessary.

2.5.3 Validating a capabilities response message

Upon receiving a capabilities response message, it MUST be validated.

A node SHOULD match the Query Identifier with a recent request that it has sent. If it is unable to match the response to a sent query then the response message SHOULD be dropped. A node MAY choose to match the source address of the response message to the destination address to which it sent the request. Note that GUE does not require addresses to be consistent in the reverse direction. A node may receive a capabilities response sourced from a different address than what it sent the request to; in that case matching the Query Identifier should be sufficient.

If the length of the last TLV exceeds the extent of the packet, then the query response message MUST be dropped. Unknown TLVs are skipped over, as are individual TLVs that have a mismatch in required length or bad data per the requirements of the TLV. TLVs may be sent in any order, may be present more than once in a packet, and the number of TLVs in a message is only limited by packet size. A receiving node may place limits on number or types of TLVs it processes.

2.5.4 Processing a capabilities response

If a valid capabilities response message is received, a node may assume that capabilities indicated in the TLVs are supported by the peer. The node can send GUE packets using those capabilities with the expectation that the peer node will process them. If a capability isn’t indicated as being supported, then a node SHOULD assume its peer doesn’t support the capability and not use it. A node MAY have other information (e.g. out of band configuration) that a peer does support a capability, in which case the capability could be used.

A capabilities query and response exchange is not a protocol negotiation, nor does it establish explicit connection-like state. The reported capabilities should be considered as advisory, and the attained information may be valid for a limited time. It is possible that a node may change its supported capabilities, may refer to a virtual IP address (VIP) where backend nodes support different capabilities, or the address for a peer is reassigned to a node that doesn’t support the same capabilities. A node MAY resend capabilities queries to a destination if it suspects that the supported capabilities might change. The echo request and reply mechanism can also be used to test that reported capabilities are supported.

3 Echo request and reply messages

This section describes the GUE echo request and echo response control messages.

3.1 Echo request

A GUE echo request message has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
   | 0 |1|   Hlen  |      0x3      |             Flags             | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ G
   |                                                               | U
   ~                  Extensions Fields (optional)                 ~ E
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ h
   |                                                               | d
   ~                    Private data (optional)                    ~ r
   |                                                               |/
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                            Data                               ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Pertinent GUE header fields are:

- C bit: Set to 1 to indicate a control message

- Proto/ctype: Set to 0x3 to indicate an echo request message

GUE flags, extension fields, and private data MAY be used in an echo request.

Control message fields are:

- Data: Contains arbitrary data set by the sender. This MAY contain an identifier to match replies with echo requests, and MAY contain a timestamp to measure round trip time.

Note that the data in an echo request is only interpretable by the sender of the echo request. A node receiving an echo request should not attempt to parse the data or interpret it, it should only reflect the data in an echo response.

3.1.1 Optional echo data format

A node MAY use the following format for the echo data. This format includes a transaction identifier, sequence number, and timestamp to facilitate matching replies to requests and measuring round trip latency.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                  Transaction identifier                       +
   |                                                               |
   +                                                               +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Sequence number                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                        Timestamp                              +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                           Data                                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fields:

- o Transaction identifier: Used to match replies with requests. This should be randomly set to a different value for each different destination

- o Sequence number: Monotomically increasing counter for sending multiple echo requests to the same node using the same the transaction identifier

- o Timestamp: Timestamp set by the echo request sender and reflected in a echo reply. Normally, this is a value taken from the system clock of the sender. The round trip latency is computed as the time the echo response was received minus the timestamp value received in the echo response. The meaning and units of the timestamp are local to the echo request sender

- o Data: Additional data that may be of relevance to the sender

3.2 Echo reply

A GUE echo reply message has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
   | 0 |1|   Hlen  |      0x4      |             Flags             | |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ G
   |                                                               | U
   ~                  Extensions Fields (optional)                 ~ E
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ h
   |                                                               | d
   ~                    Private data (optional)                    ~ r
   |                                                               |/
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                             Data                              ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Pertinent GUE header fields are:

- o C bit: Set to 1 to indicate a control message

- o Proto/ctype: Set to 0x4 to indicate an echo reply message

GUE flags, extension fields, and private data MAY be used in an echo reply.

Control message fields are:

- o Data: Reflected data that was received in an echo request

3.4 Operation

This section describes the operation of echo request and echo reply messages.

3.4.1 Sending an echo request

A node MAY send an echo request message to a peer to determine reachability or measure round trip latency. An echo request is a GUE control message that includes optional data to be reflected by a peer. A node MAY set GUE flags, extensions, and private data--particularly to test support for these as described below.

If a Transaction Identifier is used in the echo data then the sender SHOULD save it in an echo request context to match to an echo reply. The sender SHOULD set a timer to receive a response. If no response is received before timeout then the echo request context is released.

3.4.2 Receiving an echo request

When a node receives an echo request, an echo response message SHOULD be created and sent back to the source address of the echo request message. A response message SHOULD NOT set any GUE flags, extensions or private data. An exception is if the packet size exceeds MTU then the GUE fragmentation option MAY be used.

3.4.3 Receiving an echo reply

A node SHOULD match a received echo reply to an echo request that it recently sent. If the node sent a Transaction Identifier in an echo request then the value in an echo reply can be matched. Otherwise, an echo reply can be matched to a request based on the source address of the reply message matching the destination address of a recently sent request message. If sequence numbers are present they may be used to track individual echo requests and to report losses.

3.5 Testing GUE capabilities

A node MAY probe the capabilities that a GUE node supports by using the capabilities in an echo request. For instance, a node could set the remote checksum offload option in an echo request. If a corresponding echo reply is received then the node may deduce that its peer supports the feature. This mechanism can be used to verify that capabilities reported in a capabilities response are indeed supported by a peer node.

4 Security considerations

A capabilities response potentially contains detailed information about a system that might be of interest to an attacker. A node MAY choose not to respond to capabilities queries from untrusted nodes, or it may selectively curtail providing information about its capabilities.

Unsolicited capabilities response messages SHOULD NOT be accepted by a node. If a capabilities response is received, then the enclosed Query Identifier SHOULD be matched to a recent query that the node has sent. This is to prevent a attacker from spoofing someone else’s address and reporting random capabilities are supported as an attempted Denial of Service attack.

A node MAY rate limit capabilities response messages and echo reply messages to mitigate Denial of Service attacks.

5 IANA Considerations

5.1 GUE control messages

IANA is requested to assign four values in the registry for the GUE control types:

Control type	Description	Reference
0x1	Capabilities query	This document
0x2	Capabilities response	This document
0x3	Echo request	This document
0x4	Echo reply	This document

5.2 GUE capabilities TLV types

Upon publication, IANA is hereby requested to create a new registry for GUE capabilities TLV types. Initial values of this registry are as listed in Section 2.4.

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, <http://www.rfc-editor.org/info/rfc2119>.

[GUE] T. Herbert, L. Yong, and O. Zia, "Generic UDP Encapsulation" draft-ietf-intarea-gue-06

6.2. Informative References

[GUEEXTEN] Herbert, T., Yong, L., and Templin, F., "Extensions for Generic UDP Encapsulation" draft-ietf-intarea-gue-extensions-05

Author's Address

Tom Herbert
Quantonium
4701 Patrick Henry
Santa Clara, CA 95054
US

Email: tom@herbertland.com