## The TCP Echo and TCP Echo Reply Options
### draft-zimmermann-tcpm-echo-option-00

Abstract

   This document specifies the TCP Echo and TCP Echo Reply options.  It
   provides a single field a TCP sender can use to store any type of
   data that a TCP receiver simply echo unmodified back.  In contrast to
   the original TCP Echo and TCP Echo Reply options defined in RFC 1072
   the options specified in this document have slightly different
   semantics and support a variable option length.

Status of This Memo

## 1.  Introduction

This document specifies the TCP Echo and TCP Echo Reply options.  It
provides a single field a TCP sender can use to store any type of
data that a TCP receiver simply echo unmodified back.  In contrast to
the original TCP Echo and TCP Echo Reply options defined in RFC 1072
[RFC1072] the options specified in this document have a slightly
different semantics and support a variable option length.

## 2.  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 [RFC2119].  These
words only have such normative significance when in ALL CAPS, not
when in lower case.

## 3.  The TCP Echo and TCP Echo Reply options

The general structure of TCP options is defined in [RFC0793].  The
TCP Echo option is organized as indicated in Figure 1.

```
     0                   1                   2
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 ...
    +---------------+---------------+-------- ... ------+
    |    Kind A     |    Length     |    Data           |
    +---------------+---------------+-------- ... ------+
```

                   Figure 1: The TCP Echo option

The codepoint value of the TCP Echo 'Kind A' is {ToDo: Value TBA}.
The value of the 'Length' field in octets can be any value greater
than 1 as long as the TCP Echo option completely fits into TCP option
space, which may be extended (see [RFC0793], [I-D.ietf-tcpm-tcp-edo],
[I-D.briscoe-tcpm-inner-space]).  The optional 'Data' field is
available for the TCP sender to fill with any amount of any type of
data it wishes to be send back by the TCP receiver in a subsequent
TCP Echo Reply option (see Figure 2).  It is only be constrained in
size to an integer number of octets.

The TCP Echo facility is determined in both directions using a single
exchange during the 3-way handshake [RFC0793].  A TCP seeking to use
TCP Echo facility includes the TCP Echo option in the initial SYN or
SYN/ACK.  If the TCP receiver of that SYN or SYN/ACK agrees to

support TCP Echo facility, it MUST respond with TCP Echo Reply option
(see Figure 2) in its corresponding segment.

Both TCP endpoints MAY use the TCP Echo facility in any segment, but
only if the TCP Echo option was received in a segment with the SYN
bit set (i.e., SYN and SYN/ACK) or the TCP Echo Reply option was
received in response to a sent TCP Echo option.  In all cases an
endpoint MUST NOT include more than one TCP Echo option per segment.

A TCP sender MAY send an empty TCP Echo option with Length=2 on the
SYN, to only indicate that it supports the TCP Echo facility.  In
that case, the TCP receiver of that SYN MUST response with and empty
TCP Echo Reply option with Length=2 accordingly.

The TCP Echo Reply option is organized as indicated in Figure 2.

```
0                   1                   2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 ...
+---------------+---------------+-------- ... ------+
|    Kind B     |    Length     |    Data           |
+---------------+---------------+-------- ... ------+
```
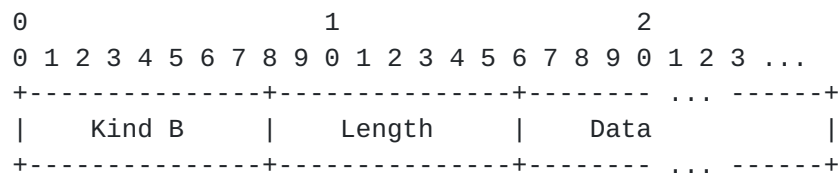
Figure 2: The TCP Echo Reply Option

A TCP receiver that does not implement the TCP Echo facility or
decides to not use the TCP Echo facility for this particular
connection MUST silently ignore any TCP Echo options it receives for
this connection.  If the TCP receiver has reflected the TCP Echo
option in its SYN/ACK during the 3-way handshake, it MUST reply to
any TCP Echo option received during this connection.

Once enabled on a connection, a TCP receiver that receives a TCP Echo
option MUST return the same bytes of the Data field in a TCP Echo
Reply option.  This TCP Echo Reply option MUST returned in the next
segment (e.g., an ACK segment) that is sent.  If due to the delayed
ACK algorithm [RFC1122] more than one TCP Echo option is received
before a reply segment is sent, the TCP receiver MUST choose only one
of the options to echo, ignoring the others; specifically, it MUST
choose the most recently received TCP Echo option to echo back (i.e.
Last In, First Out - LIFO).

4.  IANA Considerations

This specification requires IANA to allocate a value from the TCP
option kind name-space against the name

    'Kind A'
    'Kind B'

Early implementation before the IANA allocation MUST follow [RFC6994]
and use experimental option 254 and respective Experiment ID:

    0xEC01 (16 bits) for the TCP Echo option;
    0xEC02 (16 bits) for the TCP Echo Reply option;

The Echo option defined in RFC1072 [RFC1072] specifies different
semantics, which do not lend themselves for reuse.  Specifically,
RFC1072 [RFC1072] specifies to select the TCP Echo option data from
the newest segment with the oldest sequence number, while herein we
specify to return the TCP Echo option of the most recently received
segment, regardless of sequence numbers.

{ToDo: Values TBA and register them with IANA} then migrate to the
assigned option after allocation.}

5.  Security Considerations

An implementation should not rely on this facility for critical TCP
mechanisms, before ensuring that the TCP Echo option data field is
reflected back properly and unmodified.  If the TCP Echo option is
considered critical, a TCP mechanism should have means to verify the
integrity of the data contained in the TCP Echo Reply option.
Additionally, a malicious receiver or network device may infer the
utility of the data in a TCP Echo option, and interpret it for its
purposes.  A designer using the TCP Echo facility needs to consider
this, and take appropriate measures to prevent misuse of the data
sent.

Since TCP options are not delivered reliably, a TCP Echo or TCP Echo
Reply option may be lost or reordered at any time, a TCP mechanisms
MUST to deal appropriately with this occurrences.

If multiple TCP mechanisms want to make use of the TCP Echo facility,
the implementer should accommodate for that, for example by encoding
the multiple inputs accordingly into the data field of the TCP Echo
option.

Some middleboxes have been known to remove TCP options unknown to
them like those described in this document (see [Honda11]).  As the
TCP Echo and TCP Echo Reply option use two different option numbers,
it is conceivable that only one or the other may get stripped from a
segment, in one direction, resulting in an unidirectional usability
of the TCP Echo facility.

## 6.  Privacy Considerations

This document describes a new mechanism to tag individual TCP
segments.  However, the TCP options described do not expose
individual user's data.  In order to better maintain the
confidentiality of data exchanged on the wire, and to address some
aspects of security, it is NOT RECOMMENDED to send easily
decipherable data in the clear as data in the TCP Echo option.

## 7.  Acknowledgements

## 8.  References

### 8.1.  Normative References

[RFC0793]  Postel, J., "Transmission Control Protocol", STD 7, RFC
           793, September 1981.

[RFC1122]  Braden, R., "Requirements for Internet Hosts -
           Communication Layers", STD 3, RFC 1122, October 1989.

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6994]  Touch, J., "Shared Use of Experimental TCP Options", RFC
           6994, August 2013.

### 8.2.  Informative References

[Honda11]  Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A.,
           Handley, M., and H. Tokuda, "Is it still possible to
           extend TCP?", Proc. of ACM Internet Measurement Conference
           (IMC) '11, November 2011.

[I-D.briscoe-tcpm-inner-space]
           Briscoe, B., "Inner Space for TCP Options", draft-briscoe-
           tcpm-inner-space-01 (work in progress), October 2014.

[I-D.ietf-tcpm-tcp-edo]
           Touch, J. and W. Eddy, "TCP Extended Data Offset Option",
           draft-ietf-tcpm-tcp-edo-01 (work in progress), October
           2014.

   [RFC1072]   Jacobson, V. and R. Braden, "TCP extensions for long-delay
               paths", RFC 1072, October 1988.

Authors' Addresses

   Alexander Zimmermann
   NetApp, Inc.
   Sonnenallee 1
   Kirchheim  85551
   Germany

   Phone: +49 89 900594712
   Email: alexander.zimmermann@netapp.com


   Richard Scheffenegger
   NetApp, Inc.
   Am Euro Platz 2
   Vienna  1120
   Austria

   Email: rs@netapp.com


   Bob Briscoe

   Email: ietf@bobbriscoe.net
   URI:    http://bobbriscoe.net/