### TCP's Reaction to Soft Errors
### draft-ietf-tcpm-tcp-soft-errors-08.txt

Status of this Memo

Abstract

   This document describes a non-standard, but widely implemented,
   modification to TCP's handling of ICMP soft error messages, that
   rejects pending connection-requests when those error messages are
   received.  This behavior reduces the likelihood of long delays
   between connection establishment attempts that may arise in a number
   of scenarios, including one in which dual stack nodes that have IPv6
   enabled by default are deployed in IPv4 or mixed IPv4 and IPv6
   environments.

Table of Contents

## 1.  Introduction

   The handling of network failures can be separated into two different
   actions: fault isolation and fault recovery.  Fault isolation
   consists of the actions that hosts and routers take to determine that
   there is a network failure.  Fault recovery, on the other hand,
   consists of the actions that hosts and routers perform in an attempt
   to survive a network failure [RFC0816].

   In the Internet architecture, the Internet Control Message Protocol
   (ICMP) [RFC0792] is one fault isolation technique to report network
   error conditions to the hosts sending datagrams over the network.

   When a host is notified of a network error, its network stack will
   attempt to continue communications, if possible, in the presence of
   the network failure.  The fault recovery strategy may depend on the
   type of network failure taking place, and the time the error
   condition is detected.

   This document analyzes the fault recovery strategy of TCP [RFC0793],
   and the problems that may arise due to TCP's reaction to ICMP soft
   errors.  It analyzes the problems that may arise when a host tries to
   establish a TCP connection with a multihomed host for which some of
   its addresses are unreachable.  Additionally, it analyzes the
   problems that may arise in the specific scenario where dual stack
   nodes that have IPv6 enabled by default are deployed in IPv4 or mixed
   IPv4 and IPv6 environments.

   Finally, we document a modification to TCP's reaction to ICMP
   messages indicating soft errors during connection startup, that has
   been implemented in a variety of TCP/IP stacks to help overcome the
   problems outlined below.  We stress that this modification runs
   contrary to the standard behavior and this document unambiguously
   does not change the standard reaction.

   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.  Error Handling in TCP

   Network errors can be divided into soft and hard errors.  Soft errors
   are considered to be transient network failures, which are likely to
   be solved in the near term.  Hard errors, on the other hand, are
   considered to reflect network error conditions that are unlikely to
   be solved in the near future.

The Host Requirements RFC [RFC1122] states, in Section 4.2.3.9., that the ICMP messages that indicate soft errors are ICMP "Destination Unreachable" codes 0 (network unreachable), 1 (host unreachable), and 5 (source route failed), ICMP "Time Exceeded" codes 0 (time to live exceeded in transit) and 1 (fragment reassembly time exceeded), and ICMP "Parameter Problem".  Even though ICMPv6 did not exist when [RFC1122] was written, one could extrapolate the concept of soft errors to ICMPv6 "Destination Unreachable" codes 0 (no route to destination) and 3 (address unreachable), ICMPv6 "Time Exceeded" codes 0 (Hop limit exceeded in transit) and 1 (Fragment reassembly time exceeded), and ICMPv6 "Parameter Problem" codes 0 (Erroneous header field encountered), 1 (Unrecognized Next Header type encountered) and 2 (Unrecognized IPv6 option encountered) [RFC4443].

```
+--------------------------------+--------------------------------+
|              ICMP              |             ICMPv6             |
+--------------------------------+--------------------------------+
| Destination Unreachable (codes | Destination Unreachable (codes |
|          0, 1, and 5)          |            0 and 3)            |
+--------------------------------+--------------------------------+
|   Time Exceeded (codes 0 and 1)  |  Time exceeded (codes 0 and 1) |
+--------------------------------+--------------------------------+
|         Parameter Problem      | Parameter Problem (codes 0, 1, |
|                                |            and 2)              |
+--------------------------------+--------------------------------+
```

Table 1: Extrapolating the concept of soft errors to ICMPv6

When there is a network failure that is not signaled to the sending host, such as a gateway corrupting packets, TCP's fault recovery action is to repeatedly retransmit the segment until either it gets acknowledged, or the connection times out.

In the case that a host does receive an ICMP error message referring to an ongoing TCP connection, the IP layer will pass this message up to the corresponding TCP instance to raise awareness of the network failure [RFC1122].  TCP's reaction to ICMP messages will depend on the type of error being signaled.

## 2.1.  Reaction to ICMP error messages that indicate hard errors

When receiving an ICMP error message that indicates a hard error condition, TCP will simply abort the corresponding connection, regardless of the connection state.

The Host Requirements RFC [RFC1122] states, in Section 4.2.3.9, that TCP SHOULD abort connections when receiving ICMP error messages that indicate hard errors.  This policy is based on the premise that, as

hard errors indicate network error conditions that will not change in the near term, it will not be possible for TCP to usefully recover from this type of network failure.

## 2.2.  Reaction to ICMP error messages that indicate soft errors

If an ICMP error message is received that indicates a soft error, TCP will repeatedly retransmit the segment until it either gets acknowledged or the connection times out.  In addition, the TCP sender may record the information for possible later use [Stevens] (pp. 317-319).

The Host Requirements RFC [RFC1122] states, in Section 4.2.3.9, that TCP MUST NOT abort connections when receiving ICMP error messages that indicate soft errors.  This policy is based on the premise that, as soft errors are transient network failures that will hopefully be solved in the near term, one of the retransmissions will succeed.

When the connection timer expires, and an ICMP soft error message has been received before the timeout, TCP can use this information to provide the user with a more specific error message [Stevens] (pp. 317-319).

This reaction to soft errors exploits the valuable feature of the Internet that for many network failures, the network can be dynamically reconstructed without any disruption of the endpoints.

## 3.  Problems that may arise from TCP's reaction to soft errors

## 3.1.  General Discussion

Even though TCP's fault recovery strategy in the presence of soft errors allows for TCP connections to survive transient network failures, there are scenarios in which this policy may cause undesirable effects.

For example, consider a scenario in which an application on a local host is trying to communicate with a destination whose name resolves to several IP addresses.  The application on the local host will try to establish a connection with the destination host, usually cycling through the list of IP addresses, until one succeeds [RFC1123]. Suppose that some (but not all) of the addresses in the returned list are permanently unreachable.  If such a permanently unreachable address is the first in the list, the application will likely try to use the permanently unreachable address first and block waiting for a timeout before trying an alternate address.

As discussed in Section 2, this unreachability condition may or may
not be signaled to the sending host.  If the local TCP is not
signaled concerning the error condition, there is very little that
can be done other than repeatedly retransmit the SYN segment, and
wait for the existing timeout mechanism in TCP, or an application
timeout, to be triggered.  However, even if unreachability is
signaled by some intermediate router to the local TCP by means of an
ICMP soft error message, the local TCP will still repeatedly
retransmit the SYN segment until the connection timer expires (in the
hopes that the error is transient).  The Host Requirements RFC
[RFC1122] states that this timer MUST be large enough to provide
retransmission of the SYN segment for at least 3 minutes.  This would
mean that the application on the local host would spend several
minutes for each unreachable address it uses for trying to establish
the TCP connection.  These long delays between connection
establishment attempts would be inappropriate for many interactive
applications such as the web.  [Shneiderman] and [Thadani] offer some
insight into interactive systems (e.g., how the response time affects
the usability of an application).  This highlights that there is no
one definition of a "transient error" and that the level of
persistence in the face of failure represents a tradeoff.

It is worth noting that while most applications try the addresses
returned by the name-to-address function in serial, this is certainly
not the only possible approach.  For example, applications could try
multiple addresses in parallel until one succeeds, possibly avoiding
the problem of long delays between connection establishment attempts
described in this document.

## 3.2.  Problems that may arise with Dual Stack IPv6 on by Default

A particular scenario in which the above sketched type of problem may
occur regularly is that where dual stack nodes that have IPv6 enabled
by default are deployed in IPv4 or mixed IPv4 and IPv6 environments,
and the IPv6 connectivity is non-existent
[I-D.ietf-v6ops-v6onbydefault].

As discussed in [I-D.ietf-v6ops-v6onbydefault], there are two
possible variants of this scenario, which differ in whether the lack
of connectivity is signaled to the sending node, or not.

In those scenarios in which packets sent to a destination are
silently dropped and no ICMPv6 [RFC4443] errors are generated, there
is little that can be done other than waiting for the existing
connection timeout mechanism in TCP, or an application timeout, to be
triggered.

In scenarios where a node has no default routers and Neighbor

Unreachability Detection (NUD) [RFC4861] fails for destinations
assumed to be on-link, or where firewalls or other systems that
enforce scope boundaries send ICMPv6 errors, the sending node will be
signaled of the unreachability problem.  However, as discussed in
Section 2.2, standard TCP implementations will not abort connections
when receiving ICMP error messages that indicate soft errors.

## 4.  Deployed workarounds for long delays between connection-establishment attempts

The following subsections describe a number of workarounds for the
problem of long delays between connection-establishment attempts that
have been implemented in a variety of TCP/IP stacks.  We note that
treating soft errors as hard errors during connection establishment,
while widespread, is not part of standard TCP behavior and this
document does not change that state of affairs.  The TCPM WG
consensus was to document this widespread implementation of
nonstandard TCP behavior, but to not change the TCP standard.

### 4.1.  Context-sensitive ICMP/TCP interaction

As discussed in Section 1, it may make sense for the fault recovery
action to depend not only on the type of error being reported, but
also on the state of the connection against which the error is
reported.  For example, one could infer that when an error arrives in
response to opening a new connection, it is probably caused by
opening the connection improperly, rather than by a transient network
failure [RFC0816].

A number of TCP implementations have modified their reaction to soft
errors, to treat the errors as hard errors in the SYN-SENT or SYN-
RECEIVED states.  For example, this workaround has been implemented,
for example, in the Linux kernel since version 2.0.0 (released in
1996) [Linux].  However, it should be noted that this change violates
section 4.2.3.9 of [RFC1122], which states that these Unreachable
messages indicate soft error conditions and therefore TCP MUST NOT
abort the corresponding connection.

[RFC3168] states that a host that receives a RST in response to the
transmission of an ECN-setup SYN packet MAY resend a SYN with CWR and
ECE cleared.  This is meant to deal with faulty middle-boxes that
reject connections when a SYN segment has the ECE and CWR bits set.
Given that this section describes a modification that processes ICMP
error messages as hard errors when they are received for a connection
in any of the non-synchronized states, systems implementing this
behavior could resend the SYN segment with the ECE and CWR bits
cleared when an ICMP error message is received in response to a SYN

segment that had these bits set.

Section 4.2 discusses a more conservative approach than that sketched
above, that is implemented in FreeBSD.

## 4.2.  Context-sensitive ICMP/TCP interaction with repeated confirmation

A more conservative approach than simply treating soft errors as hard
errors as described above would be to abort a connection in the SYN-
SENT or SYN-RECEIVED states only after an ICMP Destination
Unreachable has been received a specified number of times, and the
SYN segment has been retransmitted more than some specified number of
times.

Two new parameters would have to be introduced to TCP, to be used
only during the connection-establishment phase: MAXSYNREXMIT and
MAXSOFTERROR.  MAXSYNREXMIT would specify the number of times the SYN
segment would have to be retransmitted before a connection is
aborted.  MAXSOFTERROR would specify the number of ICMP messages
indicating soft errors that would have to be received before a
connection is aborted.

Two additional state variables would need to be introduced to store
additional state information during the connection-establishment
phase: "nsynrexmit" and "nsofterror".  Both would be initialized to
zero when a connection attempt is initiated, with "nsynrexmit" being
incremented by one every time the SYN segment is retransmitted and
"nsofterror" being incremented by one every time an ICMP message that
indicates a soft error is received.

A connection in the SYN-SENT or SYN-RECEIVED states would be aborted
if "nsynrexmit" was greater than MAXSYNREXMIT and "nsofterror" was
simultaneously greater than MAXSOFTERROR.

This approach would give the network more time to solve the
connectivity problem than simply aborting a connection attempt upon
reception of the first soft error.  However, it should be noted that
depending on the values chosen for the MAXSYNREXMIT and MAXSOFTERROR
parameters, this approach could still lead to long delays between
connection establishment attempts, thus not solving the problem.  For
example, BSD systems abort connections in the SYN-SENT or the SYN-
RECEIVED state when a second ICMP error is received, and the SYN
segment has been retransmitted more than three times.  They also set
up a "connection-establishment timer" that imposes an upper limit on
the time the connection establishment attempt has to succeed, which
expires after 75 seconds [Stevens2] (pp. 828-829).  Even when this
policy may be better than the three-minutes timeout policy specified
in [RFC1122], it may still be inappropriate for handling the

potential problems described in this document.  This more
conservative approach has been implemented in BSD systems for more
than ten years [Stevens2].

We also note that the approach given in this section is a generalized
version of the approach sketched in the previous section.  In
particular, with MAXSOFTERROR set to 1 and MAXSYNREXMIT set to zero
the schemes are identical.


## 5.  Possible drawbacks of changing ICMP semantics

The following subsections discuss some of the possible drawbacks
arising from the use of the non-standard modifications to TCP's
reaction to soft errors described in Section 4.1 and Section 4.2.

### 5.1.  Non-deterministic transient network failures

In scenarios where a transient network failure affects all of the
addresses returned by the name-to-address translation function, all
destinations could be unreachable for some short period of time.  For
example, a mobile system consisting of a cell and a repeater may pass
through a tunnel, leading to a loss of connectivity at the repeater,
with the repeater sending ICMP soft errors back to the cell.  In such
scenarios, the application could quickly cycle through all the IP
addresses in the list and return an error, when it could have let TCP
retry a destination a few seconds later, when the transient problem
could have disappeared.  In this case, the modifications described
here make TCP less robust than a standards-compliant implementation.

Additionally, in many cases a domain name maps to a single IP
address.  In such a case, it might be better to try that address
persistently according to normal TCP rules, instead of just aborting
the pending connection upon receipt of an ICMP soft error.

### 5.2.  Deterministic transient network failures

There are some scenarios in which transient network failures could be
deterministic.  For example, consider a scenario in which upstream
network connectivity is triggered by network use.  That is, network
connectivity is instantiated only on an "as needed" basis.  In this
scenario, the connection triggering the upstream connectivity could
deterministically receive ICMP Destination Unreachables while the
upstream connectivity is being activated, and thus would be aborted.
Again, in this case, the modifications described here make TCP less
robust than a standards-compliant implementation.

## 5.3.  Non-compliant Network Address Translators (NATs)

   Some NATs respond to an unsolicited inbound SYN segment with an ICMP
   soft error message.  If the system sending the unsolicited SYN
   segment implements the workaround described in this document, it will
   abort the connection upon receipt of the ICMP error message, thus
   probably preventing TCP's simultaneous open through the NAT from
   succeeding.  However, it must be stressed that those NATs described
   in this section are not BEHAVE-compliant, and therefore should
   implement REQ-4 of [I-D.ietf-behave-tcp] instead.

## 6.  Security Considerations

   This document describes a non-standard modification to TCP's reaction
   to soft errors that has been implemented in a variety of TCP
   implementations.  This modification makes TCP abort a connection in
   the SYN-SENT or the SYN-RECEIVED states when it receives an ICMP
   "Destination Unreachable" message that indicates a soft error.
   Therefore, the modification could be exploited to reset valid
   connections during the connection-establishment phase.

   The non-standard workaround described in this document makes TCP more
   vulnerable to attack, even if only slightly.  However, we note that
   an attacker wishing to reset ongoing TCP connections could send any
   of the ICMP hard error messages in any connection state.

   Generally, TCP backs off its retransmission timer each time it
   retransmits the SYN segment for the same connection.  If a TCP
   implements the modification described in this document, that is,
   tries the next address in the list upon receipt of an ICMP error
   message, it might end up injecting more packets into the network than
   if it had simply retried the same address a number of times.
   However, compliant TCP implementations might already incur into this
   behaviour (e.g., as a result of cycling through the list of IP
   addressses in response to RST segments) as there are currently no
   recommendations on methods for limiting the rate at which SYN
   segments are sent for connecting to a specific destination.

   A discussion of the use of ICMP to perform a variety of attacks
   against TCP, and a number of counter-measures that minimize the
   impact of these attacks can be found in [I-D.ietf-tcpm-icmp-attacks].

   A discussion of the security issues arising from the use of ICMPv6
   can be found in [RFC4443].

7.  IANA Considerations

   This document has no actions for IANA.


8.  Acknowledgements

   The author wishes to thank Mark Allman, Ron Bonica, Ted Faber, Gorry
   Fairhurst, Sally Floyd, Tomohiro Fujisaki, Guillermo Gont, Saikat
   Guha, Alfred Hoenes, Michael Kerrisk, Eddie Kohler, Mika Liljeberg,
   Arifumi Matsumoto, Carlos Pignataro, Pasi Sarolahti, Pekka Savola,
   Pyda Srisuresh, and Joe Touch, for contributing many valuable
   comments on earlier versions of this document.

   The author wishes to express deep and heartfelt gratitude to Jorge
   Oscar Gont and Nelida Garcia, for their precious motivation and
   guidance.


9.  Contributors

   Mika Liljeberg was the first to describe how their implementation
   treated soft errors.  Based on that, the solution discussed in
   Section 4.1 was documented in [I-D.ietf-v6ops-v6onbydefault] by
   Sebastien Roy, Alain Durand and James Paugh.


10.  References

10.1.  Normative References

   [RFC0792]  Postel, J., "Internet Control Message Protocol", STD 5,
              RFC 792, September 1981.

   [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.

   [RFC1123]  Braden, R., "Requirements for Internet Hosts - Application
              and Support", STD 3, RFC 1123, October 1989.

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

   [RFC3168]  Ramakrishnan, K., Floyd, S., and D. Black, "The Addition
              of Explicit Congestion Notification (ECN) to IP",

                      RFC 3168, September 2001.

   [RFC4443]  Conta, A., Deering, S., and M. Gupta, "Internet Control
              Message Protocol (ICMPv6) for the Internet Protocol
              Version 6 (IPv6) Specification", RFC 4443, March 2006.

   [RFC4861]  Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
              "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
              September 2007.

## 10.2.  Informative References

   [I-D.ietf-behave-tcp]
              Guha, S., "NAT Behavioral Requirements for TCP",
              draft-ietf-behave-tcp-07 (work in progress), April 2007.

   [I-D.ietf-tcpm-icmp-attacks]
              Gont, F., "ICMP attacks against TCP",
              draft-ietf-tcpm-icmp-attacks-03 (work in progress),
              March 2008.

   [I-D.ietf-v6ops-v6onbydefault]
              Roy, S., Durand, A., and J. Paugh, "Issues with Dual Stack
              IPv6 on by Default", draft-ietf-v6ops-v6onbydefault-03
              (work in progress), July 2004.

   [Linux]    The Linux Project, "http://www.kernel.org".

   [RFC0816]  Clark, D., "Fault isolation and recovery", RFC 816,
              July 1982.

   [Shneiderman]
              Shneiderman, B., "Response Time and Display Rate in Human
              Performance with Computers", ACM Computing Surveys , 1984.

   [Stevens]  Stevens, W., "TCP/IP Illustrated, Volume 1: The
              Protocols", Addison-Wesley , 1994.

   [Stevens2]
              Wright, G. and W. Stevens, "TCP/IP Illustrated, Volume 2:
              The Implementation", Addison-Wesley , 1994.

   [Thadani]  Thadani, A., "Interactive User Productivity", IBM Systems
              Journal No. 1, 1981.

Appendix A.  Change log (to be removed before publication of the
             document as an RFC)

A.1.  Changes from draft-ietf-tcpm-tcp-soft-errors-07

   o  Fixes id nits.

A.2.  Changes from draft-ietf-tcpm-tcp-soft-errors-06

   o  Added a paragraph (in Section 4.1) about the interaction of the
      described modification with ECN-enabled connections

   o  Added a paragraph (in Section 6) about the possible scenario in
      which a host injects SYN segments into the network at a high rate,
      in response to ICMP soft errors.

   o  Miscellaneous editorial changes

A.3.  Changes from draft-ietf-tcpm-tcp-soft-errors-05

   o  Miscellaneous edits, clarifications, and reorganization of both
      workarounds into a single top-level section, as suggested by Pasi
      Sarolahti.

   o  Added note on non-compliant NATs, as suggested by Ted Faber and
      Saikat Guha

   o  Miscellaneous edits suggested by Gorry Fairhurst

   o  Added a table to clarify how to extrapolate the concept of ICMPv4
      "soft errors" to ICMPv6 (as suggested by Arifumi Matsumoto and
      Gorry Fairhurst).

   o  Miscellaneous edits, clarification on alternative approach by
      sending connection requests in parallel, example of mobile system
      (for non-deterministic errors), and note on the possible impact of
      the workarounds on TCP's robusteness (as suggested by Joe Touch)

A.4.  Changes from draft-ietf-tcpm-tcp-soft-errors-04

   o  Addresses feedback sent by Carlos Pignataro (adds missing error
      codes in Section 2, and fixes a number of typos/writeos).

A.5.  Changes from draft-ietf-tcpm-tcp-soft-errors-03

   o  Addresses feedback sent by Ted Faber and Gorry Fairhurst
      (miscellaneous editorial changes).

A.6.  Changes from draft-ietf-tcpm-tcp-soft-errors-02

   o  Moved appendix on FreeBSD's approach to the body of the draft.

   o  Removed rest of the appendix, as suggested by Ron Bonica and Mark
      Allman.

   o  Reworded some parts of the document to make the text more neutral.

   o  Miscellaneous editorial changes.

A.7.  Changes from draft-ietf-tcpm-tcp-soft-errors-01

   o  Addressed feedback posted by Sally Floyd (remove sentence in
      Section 2.1 regarding processing of RST segments)

A.8.  Changes from draft-ietf-tcpm-tcp-soft-errors-00

   o  Miscellaneous editorial changes

A.9.  Changes from draft-gont-tcpm-tcp-soft-errors-02

   o  Draft resubmitted as draft-ietf.

   o  Miscellaneous editorial changes

A.10.  Changes from draft-gont-tcpm-tcp-soft-errors-01

   o  Changed wording to describe the mechanism, rather than proposing
      it

   o  Miscellaneous editorial changes

A.11.  Changes from draft-gont-tcpm-tcp-soft-errors-00

   o  Added reference to the Linux implementation in Section 4.1

   o  Added Section 5

   o  Added section on Higher-Level API

   o  Added Section 4.2

   o  Moved section "Asynchronous Application Notification" to Appendix

   o  Added section on parallel connection requests

   o  Miscellaneous editorial changes

Author's Address

   Fernando Gont
   Universidad Tecnologica Nacional / Facultad Regional Haedo
   Evaristo Carriego 2644
   Haedo, Provincia de Buenos Aires  1706
   Argentina

   Phone: +54 11 4650 8472
   Email: fernando@gont.com.ar
   URI:   http://www.gont.com.ar