

TCP Maintenance and Minor
Extensions (tcpm)
Internet-Draft
Intended status: Informational
Expires: October 4, 2007

F. Gont
UTN/FRH
April 2, 2007

TCP's Reaction to Soft Errors
draft-ietf-tcpm-tcp-soft-errors-05.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on October 4, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes a non-standard, but widely implemented, modification to TCP's handling of ICMP soft error messages received in any of the non-synchronized states, that rejects connections experiencing those errors immediately. 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	3
2.	Error Handling in TCP	3
2.1.	Reaction to ICMP error messages that indicate hard errors	4
2.2.	Reaction to ICMP error messages that indicate soft errors	4
3.	Problems that may arise from TCP's reaction to soft errors . .	5
3.1.	General Discussion	5
3.2.	Problems that may arise with Dual Stack IPv6 on by Default	6
4.	A workaround for long delays between connection-establishment attempts	6
5.	A more conservative approach	7
6.	Possible drawbacks	8
6.1.	Non-deterministic transient network failures	8
6.2.	Deterministic transient network failures	8
7.	Security Considerations	8
8.	Acknowledgements	9
9.	Contributors	9
10.	References	9
10.1.	Normative References	9
10.2.	Informative References	10
Appendix A.	Change log (to be removed before publication of the document as an RFC)	10
A.1.	Changes from draft-ietf-tcpm-tcp-soft-errors-04	10
A.2.	Changes from draft-ietf-tcpm-tcp-soft-errors-03	11
A.3.	Changes from draft-ietf-tcpm-tcp-soft-errors-02	11
A.4.	Changes from draft-ietf-tcpm-tcp-soft-errors-01	11
A.5.	Changes from draft-ietf-tcpm-tcp-soft-errors-00	11
A.6.	Changes from draft-gont-tcpm-tcp-soft-errors-02	11
A.7.	Changes from draft-gont-tcpm-tcp-soft-errors-01	11
A.8.	Changes from draft-gont-tcpm-tcp-soft-errors-00	11
Author's Address	12
Intellectual Property and Copyright Statements	13

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. Among others, it analyzes the problems that may arise in scenarios where dual stack nodes that have IPv6 enabled by default are deployed in IPv4 or mixed IPv4 and IPv6 environments.

Additionally, 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.

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 didn't 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).

When there is a network failure that's 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 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 won't 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, 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 alternate addresses.

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 a 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 the interactive systems). 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.

[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) 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.](#) A workaround for long delays between connection-establishment attempts

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. However, this change violates [section 4.2.3.9 of \[RFC1122\]](#), which states that these Unreachable messages indicate soft error conditions and TCP MUST NOT abort the corresponding connection.

This workaround has been implemented, for example, in the Linux kernel since version 2.0.0 (released in 1996) [[Linux](#)]. [Section 5](#)

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

We note that the TCPM WG could not arrive at consensus on allowing the above described behavior as part of the standard. Therefore, 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.

5. A more conservative approach

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 since, at least, 1994 [[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.

[6.](#) Possible drawbacks

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](#) and [Section 5](#).

[6.1.](#) 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. In such a scenario, 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.

[6.2.](#) 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.

[7.](#) 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.

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](#)].

[8.](#) Acknowledgements

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

[9.](#) Contributors

Mika Liljeberg was the first to describe how their implementation treated soft errors. Based on that, the solution discussed in [Section 4](#) 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.

and Support", STD 3, [RFC 1123](#), October 1989.

- [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.

[10.2](#). Informative References

- [Guynes] Guynes, J., "Impact of System Response Time on State Anxiety", Communications of the ACM , 1988.
- [I-D.ietf-tcpm-icmp-attacks]
Gont, F., "ICMP attacks against TCP",
[draft-ietf-tcpm-icmp-attacks-01](#) (work in progress),
October 2006.
- [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-04](#)

Gont	Expires October 4, 2007	[Page 10]
------	-------------------------	-----------

Internet-Draft	TCP's Reaction to Soft Errors	April 2007
----------------	-------------------------------	------------

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

[A.2](#). Changes from [draft-ietf-tcpm-tcp-soft-errors-03](#)

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

[A.3](#). 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.4](#). 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.5](#). Changes from [draft-ietf-tcpm-tcp-soft-errors-00](#)

- o Miscellaneous editorial changes

[A.6](#). Changes from [draft-gont-tcpm-tcp-soft-errors-02](#)

- o Draft resubmitted as [draft-ietf](#).

- o Miscellaneous editorial changes

A.7. 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.8. Changes from [draft-gont-tcpm-tcp-soft-errors-00](#)

- o Added reference to the Linux implementation in [Section 4](#)
- o Added [Section 6](#)

Gont

Expires October 4, 2007

[Page 11]

Internet-Draft

TCP's Reaction to Soft Errors

April 2007

- o Added section on Higher-Level API
- o Added [Section 5](#)
- 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>

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).