

TCP Maintenance and Minor
Extensions (tcpm)
Internet-Draft
Expires: December 23, 2004

F. Gont
UTN/FRH
June 24, 2004

TCP's Reaction to Soft Errors
draft-gont-tcpm-tcp-soft-errors-00.txt

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

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/1id-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 December 23, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document discusses problems that may arise due to TCP's reaction to soft errors. In particular, it discusses the problem of long delays in connection establishment attempts that may arise when dual stack nodes that have IPv6 enabled by default are deployed in IPv4 or mixed IPv4 and IPv6 environments. The purpose of this document is to discuss this potential problem, and analyze the ways in which it could be worked around. It does not try to specify whether IPv6

should be enabled by default or not.

1. Introduction

The handling of network failures can be separated into two different actions: fault isolation and fault recovery. Fault isolation is the actions that hosts and routers take to determine that there is some network failure. Fault recovery, on the other hand, is the actions that hosts and routers will perform to isolate and survive a network failure.[\[8\]](#)

In the Internet architecture, the Internet Control Message Protocol (ICMP) [\[1\]](#) is used to perform the fault isolation function, that is, to report network error conditions to the hosts sending datagrams over the network.

When a host is signalled of a network error, there is still the issue of what to do to let communication survive, if possible, 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 discusses the fault recovery policy of TCP [\[2\]](#), and the problems that may arise due to TCP's policy of reaction to soft errors. In particular, it discusses the problems that arise in scenarios where dual stack nodes that have IPv6 enabled by default are deployed in IPv4 or mixed IPv4 and IPv6 environments.

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](#) [\[3\]](#).

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 will hopefully be solved in the near term. Hard errors, on the other hand, are considered to reflect permanent network conditions, which are unlikely to be solved in the near future.

Therefore, it may make sense for the fault recovery action to be different depending on the type of error being detected.

When there is a network failure that's not signalled 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 case the connection times out before the segment is acknowledged, TCP won't be able to

Gont

Expires December 23, 2004

[Page 2]

provide more information than the timeout condition.

In case a host does receive an ICMP error message about a current TCP connection, the IP layer will pass this message up to TCP to raise awareness of the network failure. [4]

TCP's reaction will depend on the type of error being signalled.

2.1 Reaction to Hard Errors

When receiving a segment with the RST bit set, or an ICMP error message indicating a hard error condition, TCP will simply abort the connection, regardless of the state the connection is in.

The "Requirements for Internet Hosts RFC -- Communication Layers" RFC [4] states, in [section 4.2.3.9](#), that TCP SHOULD abort connections when receiving ICMP errors that indicate hard errors. This policy is based on the premise that, as hard errors indicate network conditions that won't change in the near term, it will not be possible for TCP to recover from this type of network failure.

2.2 Reaction to Soft Errors

The "Requirements for Internet Hosts -- Communication Layers" RFC [4] states, in [section 4.2.3.9](#), that TCP MUST NOT abort connections when receiving ICMP errors that indicate soft errors.

If an ICMP error message is received that indicates a soft error, TCP will just record this information [9], and repeatedly retransmit the segment until either it gets acknowledged or the connection times out. 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.

In case the connection timer expires, and an ICMP error message had been received before the timeout, TCP will use this information to provide the user with a more specific error message. [9]

This handling of 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 arising 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

Gont

Expires December 23, 2004

[Page 3]

failures, there are scenarios in which this policy may cause undesirable effects.

For example, consider the case where 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 [5]. Suppose that some (but not all) of the addresses in the returned list are permanently unreachable. If they are the first IP addresses in the list, the application will try to use these addresses first.

As discussed in [Section 2](#), this unreachability condition may or may not be signalled to the sending host. If the local TCP is not signalled of the error condition, it will repeatedly retransmit the SYN segment, until the connection times out. If unreachability is signalled by some intermediate router to the local TCP by means of an ICMP error message, the local TCP will just record the error message and will still repeatedly retransmit the SYN segment until the connection timer expires. The "Requirements For Internet Hosts -- Communication Layers" RFC [4] 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 tries to use for a connection attempt. These long delays in connection establishment attempts would be inappropriate for interactive applications such as the web.

[3.2](#) Problems that arise with Dual Stack IPv6 on by Default

A scenario in which this type of problem may occur 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 [6].

As discussed in [6], there are two possible variants of this scenario, which differ in whether the lack of connectivity is signalled to the sending node, or not.

In cases where packets sent to a destination are silently dropped and no ICMPv6 [7] errors are generated, there is very little that can be done other than waiting for the existing connection timeout mechanism in TCP, or an application timeout, to be triggered.

In cases 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 signalled of

Gont

Expires December 23, 2004

[Page 4]

the unreachability problem. As discussed in [Section 2.2](#), TCP implementations will not abort connections when receiving ICMP errors that indicate soft errors. However, it would be desirable for TCP implementations to use this information to avoid the long delays in connection attempts described in [Section 3.1](#).

The following sections discuss some possible ways to solve this issue, and their potential drawbacks.

4. Possible solutions to the problem

[4.1](#) Changing TCP's reaction to soft errors

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 time 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. [\[8\]](#)

Thus, one solution is for TCP to abort a connection in the SYN-SENT or the SYN-RECEIVED states if it receives an ICMP "Destination Unreachable" message that indicates a soft error about that connection.

The "Requirements for Internet Hosts -- Communication Layers" RFC [\[4\]](#) states, in [section 4.2.3.9](#), that the ICMP "Destination Unreachable" messages that indicate soft errors are ICMP codes 0 (network unreachable), 1 (host unreachable), and 5 (source route failed). Even though ICMPv6 didn't exist when [\[4\]](#) was written, one could extrapolate the concept of soft errors to ICMPv6 Type 1 Codes 0 (no route to destination) and 3 (address unreachable).

A tangential method of handling the problem in this way would be for applications to somehow notify the TCP layer of their preference in the matter. An application could ask TCP to not abort a connection in the presence of such ICMP errors. This would allow existing TCP implementations to maintain their status quo at the expense of increased application complexity, while maintaining the reaction to "soft errors" described in this section as the "default" action.

There are drawbacks to this TCP behavior. In case there's a transient network failure affecting 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 situation, 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

been mitigated.

4.2 Asynchronous Application Notification

In [section 4.2.4.1](#), [4] states that there MUST be a mechanism for reporting soft TCP error conditions to the application. Such a mechanism (assuming one is implemented) could be used by applications to cycle through the destination IP addresses. However, this approach would require, in order to solve the potential problems described in [Section 3](#), every application to implement this logic, which would not be acceptable. Therefore, the solution described in [Section 4.1](#) should be preferred over this one.

5. Security Considerations

This document proposes to make 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" about that connection. While this could be used to reset valid connections, it must be noted that this behaviour is specified only for connections in the SYN-SENT or the SYN-RECEIVED states, and thus the window of exposure is very short. Furthermore, in order for this type to succeed, the attacker should be able to guess the four-tuple that identifies the target TCP connection. A discussion on this issue can be found in [10]. To mitigate the impact of this attack, additional constraints could be imposed in order to reset a connection upon receipt of the ICMP error. For example, the TCP sequence number of the contained in the payload of the ICMP error message could be required to be valid [2].

In any case, it must be noted that an attacker wishing to reset valid connections could perform the attack by sending any of the ICMP error messages that indicate "hard errors", not only for connections in the SYN-SENT or the SYN-RECEIVED states, but for connections in any state.

A discussion of the security issues arising from the use of ICMPv6 can be found in [7].

6. Acknowledgements

The author wishes to thank Michael Kerrisk, Mika Liljeberg, Pasi Sarolahti, and Pekka Savola, for contributing many valuable comments.

7. Contributors

Mika Liljeberg was the first to describe how their implementation treated soft errors. Based on that, the solutions discussed in

Gont

Expires December 23, 2004

[Page 6]

[Section 4](#) were documented in [\[6\]](#) by Sebastien Roy, Alain Durand and James Paugh.

8. References

[8.1](#) Normative References

- [1] Postel, J., "Internet Control Message Protocol", STD 5, [RFC 792](#), September 1981.
- [2] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [4] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, [RFC 1122](#), October 1989.
- [5] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), October 1989.
- [6] Roy, S., Durand, A. and J. Paugh, "Issues with Dual Stack IPv6 on by Default", [draft-ietf-v6ops-v6onbydefault-02](#) (work in progress), May 2004.
- [7] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 2463](#), December 1998.

[8.2](#) Informative References

- [8] Clark, D., "Fault isolation and recovery", [RFC 816](#), July 1982.
- [9] "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley , 1994.
- [10] "Slipping in the Window: TCP Reset Attacks", 2004 CanSecWest Conference , 2004.

Author's Address

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

Phone: +54 11 4650 8472
EMail: fernando@gont.com.ar

Intellectual Property Statement

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.

Disclaimer of Validity

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 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.

Copyright Statement

Copyright (C) The Internet Society (2004). 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.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

