

TCP Maintenance and Minor
Extensions (tcpm)
Internet-Draft
Expires: January 10, 2005

L. Eggert
NEC
July 12, 2004

TCP Abort Timeout Option
draft-eggert-tcpm-tcp-abort-timeout-option-01

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/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 January 10, 2005.

Copyright Notice

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

Abstract

The TCP Abort Timeout Option allows conforming TCP implementations to exchange requests for individual, per-connection abort timeouts. The TCP abort timeout controls how long transmitted data may remain unacknowledged before a connection is aborted. TCP implementations typically use a single, system-wide timeout value. Using individual, per-connection timeouts allows established TCP connections to survive extended periods of disconnection.

Eggert

Expires January 10, 2005

[Page 1]

Internet-Draft

TCP Abort Timeout Option

July 2004

1. Introduction

The Transmission Control Protocol (TCP) specification [1] includes a "user timeout" that defines the maximum amount of time that transmitted segments may remain unacknowledged before TCP will abort the connection. If a disconnection lasts longer than the user timeout, no acknowledgments are received for any transmission attempt, including keep-alives [5], and the TCP connection will abort when the user timeout occurs.

The TCP specification [1] does not constrain the permitted values for user timeouts. Many TCP implementations default to user timeout values of a few minutes [5]. Instead of a single user timeout, some TCP implementations offer finer-grained mechanisms. For example, Solaris supports different timeouts depending on whether a TCP connection is in the SYN-SENT, SYN-RECEIVED, or ESTABLISHED state [6]. (The host requirements RFC [2] mandates a timeout of at least three minutes for the SYN-SENT case.)

System-wide user timeouts are a useful basic mechanism. However, the ability to selectively choose individual user timeout values for different connections can improve TCP operation in scenarios that are currently not well supported.

Mobile hosts that change network attachment points based on current location are one example. Such hosts, maybe using MobileIP [7] or HIP [8], are only intermittently connected to the Internet. In between connected periods, mobile hosts may experience periods of disconnection during which no network service is available [9][10][11]. Other factors that can cause disconnections are high levels of transient congestion and link or routing failures inside the network.

In scenarios similar to the ones described above, a host may not know exactly when or for how long it will be disconnected from the network, but it might expect such events due to past mobility patterns and thus benefit from using longer abort timeouts. In other scenarios, the length and time of a network disconnection may even be predictable. For example, an orbiting node on a satellite experiences disconnections due to line-of-sight blocking by other planetary bodies. The disconnection times and durations of such a host may be easily computable from orbital mechanics.

In these examples above, as well as other cases, established TCP connections between two peers can abort when a disconnection exceeds the system-wide default user timeout. This document specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to exchange per-connection abort timeout requests. This allows, for

example, mobile hosts to maintain TCP connections across disconnected periods that are longer than their system's default user timeout. A second use of the TCP Abort Timeout Option is exchange of shorter-than-default abort timeouts. This can allow busy servers to explicitly notify their clients that they will maintain the state associated with established connections only across short periods of disconnection.

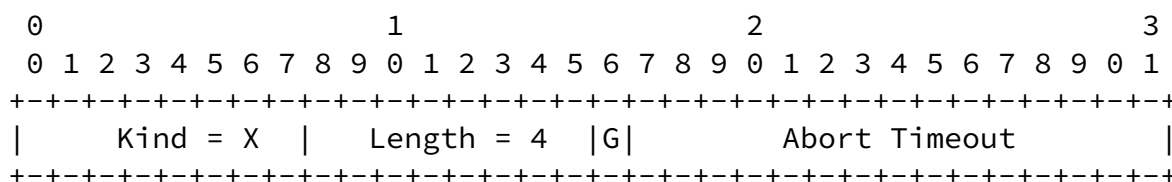
TCP Abort Timeout Options allow hosts to both request specific abort timeouts for new connections and to request changes to the effective abort timeouts of established connections. The latter allows connections to start with short timeouts and only request longer

timeouts when disconnection was imminent, and only for connections considered important. The ability to request changes to abort timeouts of established connections is also useful to raise the abort timeout after in-band authentication has occurred. For example, peers could request longer abort timeouts for the TCP connections underlying two-way authenticated TLS connections [12] after their authentication handshakes.

2. Conventions

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].

3. Specification



(One tick mark represents one bit.)

Figure 1: Format of the TCP Abort timeout Option

Figure 1 shows the format of the TCP Abort Timeout Option. It contains these fields:

Kind (8 bits):

A TCP option number [1] to be assigned by IANA upon publication of this document (see Section 5.)

Length (8 bits):

Length of the TCP option in octets [1]; its value MUST be 4.

Granularity (1 bit):

Granularity bit, indicating the granularity of the "Abort Timeout" field. When set ($G = 1$), the time interval in the "Abort Timeout" field **MUST** be interpreted as minutes. Otherwise ($G = 0$), the time interval in the "Abort Timeout" field **MUST** be interpreted as seconds.

Abort Timeout (15 bits):

Specifies the abort timeout suggestion for this connection. It **MUST** be interpreted as a 15-bit unsigned integer. The granularity of the timeout (minutes or seconds) depends on the "G" field.

[3.1](#) Operation

Sending a TCP Abort Timeout Option signals to the receiving peer that the sender will start to use the indicated abort timeout value locally for the connection and is requesting that the receiving peer should start to use a corresponding abort timeout for it. [Section 3.2](#) discusses the effects of different timeout values.

When a host that supports the TCP Abort Timeout Option receives one, it decides whether to change the connection's local abort timeout accordingly. Generally, hosts **SHOULD** honor requests for changes to the abort timeout, unless security concerns or external policies indicate otherwise (see [Section 4](#).) If so, hosts **MAY** ignore incoming TCP Abort Timeout Options and **MAY** use a different abort timeout for the connection.

A TCP Abort Timeout Option with a value of zero (i.e., "now") is nonsensical and **MUST NOT** be sent. If received, it **MUST** be ignored. [Section 3.2](#) discusses potentially problematic effects of other abort timeout durations.

Hosts **SHOULD** impose upper and lower limits on the abort timeouts they use. [Section 3.2](#) discusses abort timeout limits.

The abort timeout value included in a TCP Abort Timeout Option specifies the requested abort timeout during a connection's

synchronized states (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, or LAST-ACK.) Connections in other states MUST use standard timeout values [\[1\]](#)[\[2\]](#).

(NB: A future version of this document may extend per-connection abort timeouts to the SYN-SENT and SYN-RECEIVED states in a way that conforms to the required minimum timeouts.)

A TCP implementation that does not support the TCP Abort Timeout

Eggert

Expires January 10, 2005

[Page 4]

Internet-Draft

TCP Abort Timeout Option

July 2004

Option SHOULD silently ignore it [\[2\]](#), thus ensuring interoperability.

It is important to note that TCP Abort Timeout Options do not change the semantics of the TCP protocol. Hosts remain free to abort connections at any time for any reason, whether or not they use custom abort timeouts or have requested the peer to use them. Connections may also terminate due to other reasons, such as stateful firewalls that terminate connections after apparent periods of idleness.

[3.1.1](#) Operation during the SYN Handshake

A host that supports the TCP Abort Timeout Option and wishes to use individual abort timeouts for a connection MUST include an appropriate TCP Abort Timeout Option in its initial SYN segment.

A host that supports the TCP Abort Timeout Option MAY omit the TCP Abort Timeout Option from the initial SYN if custom abort timeouts are not required for a specific connection. It SHOULD omit the TCP Abort Timeout Option from the initial SYN if there is evidence that the peer does not support the TCP Abort Timeout Option, for example, if a prior connection attempt including a TCP Abort Timeout Option has failed.

If a host does not include a TCP Abort Timeout Option in its initial SYN, it MUST NOT include it in any other segment either and MUST ignore the contents of any received TCP Abort Timeout Option.

A host that supports the TCP Abort Timeout Option and receives a SYN segment that includes one SHOULD respond with an appropriate TCP Abort Timeout Option in its SYN-ACK segment. If an incoming SYN segment does not include a TCP Abort Timeout Option, a host MUST NOT include one in the SYN-ACK segment or any other segment either and it MUST ignore the contents of any other received TCP Abort Timeout Option.

[3.1.2](#) Operation during the Synchronized States

During the synchronized states (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, or LAST-ACK) and unless both the SYN and SYN-ACK of a connection contained TCP Abort Timeout Options, both hosts participating in the connection MUST NOT send TCP Abort Timeout Options in any other segment. Additionally, they both MUST ignore the contents of any received TCP Abort Timeout Option.

Otherwise, whenever a host changes the local abort timeout of a connection, it SHOULD include a TCP Abort Timeout Option indicating the new abort timeout in its next segment to the peer. This allows

the peer to adapt its local abort timeout for the connection accordingly.

[3.2](#) Duration of the Abort Timeout

The TCP Abort Timeout Option allows host to exchange abort timeout values from zero seconds to over 9 hours at a granularity of seconds and from zero minutes to over 22 days at a granularity of minutes.

Very short abort timeout values can affect TCP transmissions over high-delay paths. If the abort timeout occurs before an acknowledgment for an outstanding segment arrives, possibly due to packet loss, the connection aborts. Many TCP implementations default to abort timeout values of a few minutes [5]. Although the TCP Abort Timeout Option allows negotiation of short timeouts, applications requesting them should consider these effects.

Long abort timeout values allow hosts to tolerate extended periods of disconnection. However, they also require hosts to maintain the TCP state associated with connections for long periods of time. [Section 4](#) discusses the security implications of long timeout values.

To protect against these effects, implementations SHOULD impose limits on the abort timeout values they accept and use. The remainder of this section describes a RECOMMENDED scheme to limit abort timeouts based on upper and lower limits. Under the RECOMMENDED scheme to limit abort timeouts, each TCP SHOULD compute the abort timeout (USER_TIMEOUT) for a connection according to this formula:

$$\text{USER_TIMEOUT} = \min(\text{U_LIMIT}, \max(\text{LOCAL_UTO}, \text{REMOTE_UTO}, \text{L_LIMIT}))$$

Each field is to be interpreted as follows:

USER_TIMEOUT:

Resulting abort timeout value to be adopted by the local TCP for a connection.

U_LIMIT:

Current upper limit imposed on the connection's abort timeout by the local host.

L_LIMIT:

Current lower limit imposed on the connection's abort timeout by the local host.

Internet-Draft

TCP Abort Timeout Option

July 2004

LOCAL_UTO:

Current local abort timeout of the specific connection.

REMOTE_UTO:

Last received abort timeout option the peer uses for the connection, i.e., contents of the last-received TCP Abort Timeout Option.

Enforcing a lower limit (L_LIMIT) protects against connection aborts due to transient network conditions, including temporary congestion, mobility hand-offs or routing instabilities.

An upper limit (U_LIMIT) can reduce the effect of resource exhaustion attacks. [Section 4](#) discusses the details of these attacks.

Note that these limits MAY be specified as system-wide constants or at other granularities, such as on per-host, per-user or even per-connection basis. Furthermore, these limits need not be static. For example, they MAY be a function of system resource utilization or attack status and could be dynamically adapted.

The Host Requirements RFC [\[2\]](#) does not impose any limits on the length of the abort timeout. However, a time interval of at least 100 seconds is RECOMMENDED. Consequently, the lower limit (LLIMIT) SHOULD be set to at least 100 seconds when following the RECOMMENDED scheme described in this section.

[4.](#) Security Considerations

Lengthening abort timeouts has obvious security implications. Flooding attacks cause denial of service by forcing servers to commit resources for maintaining the state of throw-away connections. TCP implementations do not become more vulnerable to simple SYN flooding by implementing the TCP Abort Timeout Option, because abort timeouts negotiated during the handshake only affect the synchronized states

(ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK), which simple SYN floods never reach.

However, when an attacker completes the three-way handshakes of its throw-away connections it can amplify the effects of resource exhaustion attacks, because the attacked server must maintain the connection state associated with the throw-away connections for longer durations. Because connection state is kept longer, lower-frequency attack traffic, which may be more difficult to detect, can already cause resource exhaustion.

Several approaches can help mitigate this issue. First, implementations can require prior peer authentication, e.g., using

IPsec [[13](#)], before accepting long abort timeouts for the peer's connections. Similarly, a host can only start to accept long abort timeouts for an established connection after in-band authentication has occurred, for example, after a TLS handshake across the connection has succeeded [[12](#)]. Although these are arguably the most complete solutions, they depend on external mechanisms to establish a trust relationship.

A second alternative that does not depend on external mechanisms would introduce a per-peer limit on the number of connections that may use increased abort timeouts. Several variants of this approach are possible, such as fixed limits or shortening accepted abort timeouts with a rising number of connections. Although this alternative does not eliminate resource exhaustion attacks from a single peer, it can limit their effects.

Per-peer limits cannot protect against distributed denial of service attacks, where multiple clients coordinate a resource exhaustion attack that uses long abort timeouts. To protect against such attacks, TCP implementations could reduce the duration of accepted abort timeouts with increasing resource utilization.

TCP implementations under attack may be forced to shed load by resetting established connections. Some load-shedding heuristics, such as resetting connections with long idle times first, can negatively affect service for intermittently connected, trusted peers that have negotiated long abort timeouts. On the other hand, resetting connections to untrusted peers that use long abort timeouts may be effective. In general, using the peers' level of trust as a parameter during the load-shedding decision process may be useful.

Finally, upper and lower limits on abort timeouts, discussed in [Section 3.2](#), can be an effective tool to limit the impact of these sorts of attacks.

[5.](#) IANA Considerations

This section is to be interpreted according to [\[4\]](#).

This document does not define any new namespaces. It uses an 8-bit TCP option number maintained by IANA at <http://www.iana.org/assignments/tcp-parameters>.

[6.](#) Acknowledgments

This revision of the document incorporates several ideas from Fernando Gont's "Adaptive User Timeout" mechanism [\[14\]](#) that is based on the -00 revision of this document. The two documents are

| | | |
|--------|--------------------------|----------|
| Eggert | Expires January 10, 2005 | [Page 8] |
|--------|--------------------------|----------|

| | | |
|----------------|--------------------------|-----------|
| Internet-Draft | TCP Abort Timeout Option | July 2004 |
|----------------|--------------------------|-----------|

currently being merged, but a few issues remain to be resolved. In the meantime, this revision documents the current state of the "abort timeout" proposal.

The following people have improved this document through thoughtful

suggestions: Mark Allmann, Marcus Brunner, Wesley Eddy, Tom Henderson, Joseph Ishac, Michael Kerrisk, Kostas Pentikousis, Juergen Quittek, Stefan Schmid, Simon Schuetz, and Martin Stiernerling.

[7.](#) References

[7.1](#) Normative References

- [1] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [2] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, [RFC 1122](#), October 1989.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [4] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.

[7.2](#) Informative References

- [5] "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley , 1994.
- [6] Sun Microsystems, "Solaris Tunable Parameters Reference Manual", Part No. 806-7009-10, 2002.
- [7] Perkins, C., "IP Mobility Support for IPv4", [RFC 3344](#), August 2002.
- [8] Moskowitz, R., "Host Identity Protocol Architecture", [draft-moskowitz-hip-arch-05](#) (work in progress), October 2003.
- [9] Schuetz, S., "Network Support for Intermittently Connected Mobile Nodes", M.S. Thesis, University of Mannheim, Germany, June 2004.

[10] Schuetz, S., Eggert, L., Schmid, S. and M. Brunner, "Protocol Enhancements for Intermittently Connected Hosts", under submission (work in progress), July 2004.

[11] Ott, J. and D. Kutscher, "Drive-Thru Internet: IEEE 802.11b for

Eggert

Expires January 10, 2005

[Page 9]

Internet-Draft

TCP Abort Timeout Option

July 2004

Automobile Users", Proc. INFOCOM 2004, March 2004.

[12] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", [RFC 2246](#), January 1999.

[13] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.

[14] Gont, F., "TCP Adaptive User TimeOut (AUTO) Option", [draft-gont-tcpm-tcp-auto-option-00](#) (work in progress), May 2004.

[15] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.

Author's Address

Lars Eggert
NEC Network Laboratories
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 90511 43

Fax: +49 6221 90511 55
EMail: lars.eggert@netlab.nec.de
URI: <http://www.netlab.nec.de/>

[Appendix A](#). Document Revision History

| Revision | Comments |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00 | Initial version. |
| 01 | Updated draft based on WG feedback. Also incorporated some ideas from Fernando Gont's "Adaptive User Timeout" proposal [14] that is based on the -00 revision of this document. |

Eggert

Expires January 10, 2005

[Page 10]

Internet-Draft

TCP Abort Timeout Option

July 2004

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.