Network Working Group
Internet-Draft
Expires: October 13, 2004

L. Eggert
NEC

April 14, 2004

TCP Abort Timeout Option
draft-eggert-tcpm-tcp-abort-timeout-option-00

Status of this Memo

Copyright Notice

Abstract

The TCP Abort Timeout Option allows conforming TCP implementations to

negotiate individual, per-connection abort timeouts. Lengthening abort timeouts allows established TCP connections to survive periods of disconnection.

1. Introduction

Some hosts are only intermittently connected to the Internet. One example is mobile hosts that change network attachment points based on current location, for example using MobileIP [5] or HIP [6]. In between connected periods, mobile hosts may experience disconnected periods during which no network service is available. When such hosts use the Transmission Control Protocol (TCP) [1], their established TCP connections can abort during periods of disconnection.

The TCP specification [1] includes a "user timeout" that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort. 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 [7].

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 [8]. (The Host Requirements document [2] requires the timeout to be at least three minutes for the SYN-SENT case.)

This document specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts. This allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system's default abort timeout.

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. Specification

Figure 1 shows the format of the TCP Abort Timeout Option. In Figure 1, "X" is a TCP option number to be assigned by IANA upon publication of this document (see Section 5.) "Abort Timeout" is the desired abort timeout of the connection, specified in seconds.

```
                             +----------+----------+
                             |  Kind=X  | Length=6 |
          +----------+----------+----------+----------+
          |              Abort Timeout                |
          +----------+----------+----------+----------+
```

Figure 1: TCP Abort Timeout Option

2.1 Operation

A host wishing to negotiate a specific abort timeout for a connection MAY include the TCP Abort Timeout Option in any segment with a SYN flag, i.e., either the initial SYN or the SYN-ACK. It MUST NOT include an Abort Timeout Option in any other segment.

The timeout value included in the option specifies the proposed abort timeout for the connection. Connections use abort timeouts negotiated with Abort Timeout Options during the ESTABLISHED state only. When connections are in other states, normal timeouts are used [1][2].

A host proposing an abort timeout to its peer MUST be prepared to accept a shorter timeout value than proposed after the negotiation. See Section 2.2 for a discussion of valid timeout values.

Upon receipt of a segment with the Abort Timeout Option, the receiving host decides whether to accept, shorten, or reject its peer's proposed abort timeout. Section 2.3 discusses the specifics of this decision.

When a receiving host accepts or shortens the offered abort timeout, it MUST include an Abort Timeout Option with the corresponding timeout value in the next segment it sends. This will either be the SYN-ACK, if it received the peer's Timeout Abort Option in the SYN segment, or the first ACK if it received the option in the SYN-ACK segment.

This specification allows both the initiator of a TCP connection (i.e., the node sending the SYN) as well as the responder of a TCP connection (i.e., the node receiving the SYN) to initiate an abort timeout negotiation during the connection's three-way handshake. Figure 2 illustrates the two allowed exchanges.

```
   Initiator         Responder             Initiator         Responder
       |                 |                     |                 |
       |     SYN+ATO     |                     |       SYN       |
       |---------------->|                     |---------------->|
       |                 |                     |                 |
       |   SYN/ACK+ATO   |                     |   SYN/ACK+ATO   |
       |<----------------|                     |<----------------|
       |                 |                     |                 |
       |       ACK       |                     |     ACK+ATO     |
       |---------------->|                     |---------------->|
       |                 |                     |                 |
       V                 V                     V                 V
```

Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

If the receiving host accepts the peer's offered abort timeout, it MUST echo the offered timeout value in the Abort Timeout Option it sends. If it shortens the timeout, it MUST send an Abort Timeout Option with a timeout value that is correspondingly less than the offered one. In both cases, it MUST then use the selected abort timeout value for the connection.

If the receiving host decides to reject its peer's offered abort timeout, it MUST NOT include an Abort Timeout Option in the next segment it sends. It MUST then also use its default timeout value for the connection.

Whether it accepts, shortens or rejects the peer's offer, the receiving host MUST NOT include an Abort Timeout Option in any other segment.

The host that initially proposed the Abort Timeout Option analyzes the next segment it receives from its peer. If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option. This can be a different abort timeout than initially proposed, if the peer decided to shorten it.

A TCP implementation that does not support the TCP Abort Timeout Option SHOULD silently ignore it [2]. This causes the connection to use the default abort timeout, thus ensuring interoperability. For the same reason, TCP implementations that support the TCP Abort Timeout Option SHOULD ignore it when they receive the option in a segment other than a SYN or first ACK. Future extensions to this document may modify the latter rule (see Section 3.)

2.2 Length of the Abort Timeout

The TCP specification [1] does not define a range for permitted abort timeouts. Similarly, this document does not restrict the range of timeout values used with the TCP Abort Timeout Option. The 32-bit value in the option can express abort timeouts from zero seconds to over 136 years.

Very short timeout values can affect TCP retransmissions over high-delay paths. Many TCP implementations default to abort timeout values of a few minutes [7]. Although the TCP Abort Timeout Option allows negotiation of shorter timeouts, applications requesting such short timeouts 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.

2.3 Accepting, Shortening or Rejecting an Abort Timeout Option

As described in Section 2, when a host receives a TCP Abort Timeout Option from its peer, it may accept, shorten, or reject the offer. To accept the offer, the host echoes the proposed value back to its peer by including a TCP Abort Timeout Option in its next segment. To shorten the offer, it lowers the timeout value accordingly before sending. To reject the offer, it does not include such an option in its next segment.

The decision of whether to accept, shorten, or reject an offered timeout is a local policy decision. This document does not further discuss this decision, other than discussing relevant security considerations in Section 4.

3. Future Extensions and Open Questions

This section discusses possible future extensions or modifications of the TCP Abort Timeout Option described in this document. Community input on these items is highly appreciated.

3.1 Timeout Negotiation for Established Connections

The processing of the Abort Timeout Option defined in Section 2 requires timeout negotiation to occur during a connection's three-way handshake. Although this simplifies the protocol, it eliminates the possibility of negotiating new timeouts after connection establishment.

Negotiation of timeouts for established connections may be useful. Connections could default to starting with shorter timeouts and only negotiate longer timeouts when disconnection was imminent. This may reduce the amount of state held during times of disconnection. A future revision of this document may specify means for negotiating timeouts for established connections.

3.2 Abort Timeout Option Granularity and Length

Currently, the Abort Timeout Option specifies abort timeouts as 32-bit values with a granularity of seconds (Section 2.) Consequently, the current option format can express aborts timeouts from zero seconds to over 136 years.

It may be useful to permit finer-grained timeouts, e.g., milliseconds instead of seconds. Likewise, it may be useful to lengthen or shorten the timeout field in the option to permit longer timeouts or reduce the required header space. For example, a 16-bit timeout value with a granularity of seconds allows timeout values up to 18.2 hours, whereas a 32-bit timestamp with millisecond granularity allows timeout values up to 49.7 days.

3.3 Upper and Lower Bounds on Abort Timeouts

With the current Abort Timeout Option, initiators can propose abort timeouts and recipients may reduce the offered timeout values. This scheme grants recipients some level of control over abort timeouts for their connections. When initiators propose very long timeouts, recipients may reduce the timeout offer to an acceptable length that may still be longer than the default. In a sense, timeouts offered by initiators are upper bounds on the actual timeouts used for established connections.

The current Abort Timeout negotiation does not permit initiators to specify a corresponding lower bound. Recipients may arbitrarily shorten timeout offers, potentially resulting in much shorter timeouts than initiators desired. A straightforward extension of the current TCP Timeout Option would offer both upper and lower bounds for user timeouts. Recipients of the extended offer would then choose a specific timeout within the offered bounds.

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 Abort Timeout Option, because abort timeouts negotiated during the handshake only affect the ESTABLISHED state, 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 [9], before accepting long abort timeouts for the peer's connections. Although this is arguably the most complete solution, it depends on external mechanisms to establish trust.

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 length 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.

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 in http://www.iana.org/assignments/tcp-parameters.

IANA is requested to assign a TCP option number upon publication of this document.

6. Acknowledgments

The following people have improved this document through thoughtful suggestions: Simon Schuetz, Stefan Schmid, Martin Stiemerling, and Marcus Brunner.

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] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

[6] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-05 (work in progress), October 2003.

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

[8] Sun Microsystems, "Solaris Tunable Parameters Reference Manual", Part No. 806-7009-10, 2002.

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

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.

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 IETF's procedures with respect to rights in IETF 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

Acknowledgment

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