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

L. Eggert
NEC
F. Gont
UTN/FRH
July 12, 2004

TCP User TimeOut (UTO) Option
draft-eggert-gont-tcpm-tcp-uto-option-00.txt

Status of this Memo

Copyright Notice

Abstract

The original TCP specification (RFC793) defines a "USER TIMEOUT" parameter that sets the policy as to when a user connection should be aborted. However, TCP provides no means of letting users suggest an abort policy to a remote peer dynamically. Even though a fixed

policy may work well in many cases, there are a number of scenarios where a fixed USER TIMEOUT value may be inappropriate, and some means of setting the abort policy dynamically may be necessary for TCP to be used effectively in such scenarios. This document defines a new TCP option, which lets a TCP peer suggest a USER TIMEOUT value to a remote TCP during the connection-establishment phase, and modify it during the life of a connection, thus adapting TCP's connection-abort policy as necessary.

Table of Contents

1. Introduction

The original TCP specification [1] defines a USER TIMEOUT parameter, which sets the policy as to when a connection should be aborted. This parameter is usually set on a per-system basis, and there is no way for a TCP to suggest a value of USER TIMEOUT to be used for a connection by a remote peer.

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 policies. For example, Solaris supports different timeouts depending on whether a TCP connection is in the SYN-SENT, SYN-RECEIVED, or ESTABLISHED state [6].

Even though having such a fixed policy may work well in many cases, there are scenarios in which the default USER TIMEOUT may be inappropriate. These scenarios include, but are not limeted to, the following:

- A mobile host connected to a network by means of a wireless link may move through locations where its network connectivity is interrupted by physical surroundings or lack of infrastructure.

- High levels of congestion may develope during the life of a connection, resulting in transient periods of disconnection.

In such cases, valid connections may be aborted due to an incorrect abort policy.

There are scenarios in which a host may not know exactly when or for how long it may experience network disconnection, but due to its mobility pattern, it might expect such events, and thus benefit from using a longer USER TIMEOUT to mitigate their impact. For example, as a mobile smartphone moves between cells of coverage, its motion and connectivity may be unpredictable in the short term. In the

long-term, however, its network connectivity may be expected to return.

There are scenarios in which the length of time a host will experience network disconnection is even predictable. For example, if an orbiting node on a satellite experiences disconnections due to line-of-sight blocking by other bodies, the duration of the disconnection periods may be easily computable from orbital mechanics.

In the described scenarios, as well as in other ones, established TCP connections between two peers may be aborted if a disconnection exceeds the system-wide default USER TIMEOUT.

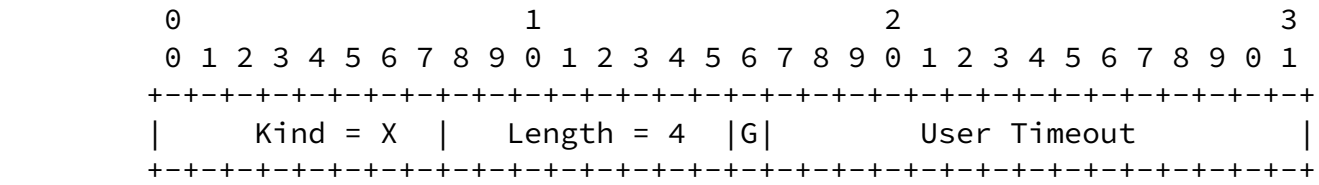
This document defines a new TCP option, that lets TCP implementations suggest a USER TIMEOUT value during the connection-establishment phase, and modify it during the life of a connection, thus adapting TCP's connection-abort policy as necessary.

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 [2].

3. Specification

3.1 Option Format



Note that one tick mark represents one bit position

Figure 1: User Timeout (UTO) Option Format

Each field is to be interpreted as follows:

Kind (8 bits):

This is the "Kind" field as specified in [1]. The "X" in Figure 1 is an option number to be assigned by IANA upon publication of this document (see Section 7).

Length (8 bits):
Length of the TCP option in octets [1]; its value MUST be 4.

G (1 bit):
This is the "Granularity" bit. It indicates the granularity of the "User Timeout" field. When set (G = 1), the time interval in the "User Timeout" field MUST be interpreted as being specified in minutes. Otherwise (G = 0), the time interval in the "User Timeout" field MUST be interpreted as being specified in seconds.

User Timeout (15 bits):
This field, together with the Granularity bit, specifies the USER TIMEOUT suggested by the remote peer for this connection. It MUST be interpreted as a 15-bit unsigned integer. The units of this field are specified by the "G" bit.

3.2 Operation

A TCP implementation that supports the User TimeOut (UTO) Option MUST set this option when performing a connection request (i.e., in its initial SYN segment) to indicate that the option is supported, and to suggest a USER TIMEOUT value to be used for the connection.

A TCP implementation that supports the TCP User TimeOut Option and receives a SYN segment that includes one SHOULD include a TCP User TimeOut Option in its SYN-ACK segment. If an incoming SYN segment does not include a TCP User TimeOut Option, the local TCP MUST NOT include the UTO option in the SYN-ACK segment nor in any other segment, and MUST ignore any TCP User Timeout Option received during the life of the connection.

A TCP implementation that does not support the TCP User TimeOut Option SHOULD silently ignore it [3], thus ensuring interoperability.

A TCP MAY also use this option during the life of a connection, to suggest a new value for the USER TIMEOUT parameter, thus adapting it to the current network conditions. This could be useful in a number

of scenarios which include, but are not limited to, the following:

- A TCP that is notified of congestion by means of ECN [7] could set this option to suggest a USER TIMEOUT value that reflects the current network conditions.

- A TCP could start connections with short timeouts, and suggest longer timeouts only when disconnection is imminent.

- A TCP could start connections with short timeouts, and raise the USER TIMEOUT after in-band authentication has occurred. For example, TCP peers could suggest longer USER TIMEOUT values for TCP connections for which a TLS handshake across the connection has succeeded [8].

The setting of this option means "I suggest we use a USER TIMEOUT of X". The value of "X" may be larger or smaller than the default USER TIMEOUT (see Section 4).

Hosts SHOULD impose upper and lower limits on the USER TIMEOUT. A discussion of these limits can be found in Section 5.

Each TCP will adopt a USER TIMEOUT as defined by equation (1):


```
USER_TIMEOUT = min( U_LIMIT, max(LOCAL_UTO, REMOTE_UTO, L_LIMIT))
```

Equation 1: USER TIMEOUT to be adopted for the connection

Each field is to be interpreted as follows:

USER_TIMEOUT:
: USER TIMEOUT value to be adopted by the local TCP for this connection.

U_LIMIT:
: Current upper limit imposed by this host on the USER TIMEOUT of this connection.

L_LIMIT:
Current lower limit imposed by this host on the USER TIMEOUT of this connection.

LOCAL_UTO:
The "USER TIMEOUT" value suggested for this connection by the local TCP, by means of the UTO Option.

REMOTE_UTO:
Last "USER TIMEOUT" value suggested by the remote TCP peer by means of the UTO Option.

The adopted USER TIMEOUT SHOULD be used only for connections that are in one of the synchronized states (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK or TIME-WAIT). Connections in other states MUST use the default USER TIMEOUT values [1][3].

Note that the USER TIMEOUT is not negotiated in any way. Each peer just "suggests" what USER TIMEOUT should be adopted for the connection. As can be inferred from the equation above, each peer may end up adopting a different timeout value.

It is important to note that TCP User 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 user timeouts or have suggested the peer to use them.

4. Range of valid values

The User TimeOut Option allows a TCP peer to suggest USER 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. However, implementations SHOULD impose limits on the USER TIMEOUT

values actually adopted. A discussion of these limits can be found in Section 5.

Setting the USER TIMEOUT to very short values can affect TCP connections over high-delay paths or very unreliable links: If the user timeout occurs before an acknowledgement for an outstanding segment arrives, possibly due to packet loss, the connection is

aborted. Although the TCP User TimeOut Option allows the use of short timeouts, applications suggesting them should consider these effects.

Long USER 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 8 discusses the security implications of long USER TIMEOUT values.

5. System limits on the USER TIMEOUT

Implementations SHOULD impose an upper limit (U_LIMIT) and a lower limit (L_LIMIT) on the value of the USER TIMEOUT.

A lower limit gives some minimum tolerance for changing network conditions, mobility periods that take a host completely off the network for some time, and time for convergence of binding updates when a host moves.

An upper limit helps prevent attacks in which resources are exhausted by creating connections to the target host and then throwing them away without signalling this to the attacked host's TCP. (See Section 8). An upper limit could also help to save resources on a busy server.

Note that these limits MAY be set, for example, on a per-host or per-user basis. Furthermore, these limits need not be fixed. For example, they MAY be a function of the system resources that are available when the USER TIMEOUT is to be selected for a connection.

The Host Requirements RFC [3] does not impose any limits for the USER TIMEOUT. However, a time interval of at least 100 seconds is RECOMMENDED. Thus, the lower limit (L_LIMIT) SHOULD be set to at least 100 seconds.

A TCP User Timeout Option with a value of zero (i.e., "now") is nonsensical and MUST NOT be sent. If received, it MUST be ignored.

6. Interoperability issues

6.1 Firewalls

Stateful firewalls are known to reset connections after some fixed period of inactivity is detected. In case there is such a firewall between the TCP peers, then, regardless of the use of the UTO Option, connections may be lost due to the firewall policy.

6.2 TCP Keep-alive mechanism

In case a TCP peer enables the TCP Keep-alive mechanism for a connection that is using the UTO Option, then the Keep-alive timer MUST be set to a value larger than that of the adopted USER TIMEOUT (specified by Equation 1).

7. 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.

8. Security Considerations

Use of the UTO Option implies that the adopted USER TIMEOUT may be larger than the default USER TIMEOUT. This could cause a host to maintain state for a connection for a longer period of time than if the default USER TIMEOUT were used. An attacker could try to exhaust resources on the target host by establishing lots of connections and aborting them without signalling this to the attacked host's TCP.

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. Similarly, a host can start to accept long abort timeouts for an established connection only after in-band authentication has occurred, for example, after a TLS handshake across the connection has succeeded [8]. 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 large user timeouts. Several variants of this approach are possible, such as fixed limits or shortening accepted user 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 user timeouts. To protect against such attacks, TCP implementations could reduce the duration of accepted user 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 user timeouts. On the other hand, resetting connections to untrusted peers that use long user timeouts may be effective. In general, using the peers' level of trust as a parameter during the load-shedding decision process may be useful.

In any case, it must be noted that the same type of attack can be performed even if the default "USER TIMEOUT" is used, since TCP requires no message exchange in order to keep a connection open. In any case, the system limits discussed in Section 5 would serve as a counter-measure against attackers trying to exploit the UTO option for this type of attack.

Note that TCP implementations do not become more vulnerable to simple SYN flooding attacks by implementing the Usert TimeOut Option, because the adopted user timeouts are used only for connections that are in one of the synchronized states (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK or TIME-WAIT) [1]. which which connections resulting from simple SYN floods never reach.

9. Acknowledgements

The following people have improved this document through thoughtful suggestions: Mark Allmann, Marcus Brunner, Wesley Eddy, Ted Faber, Guillermo Gont, Tom Henderson, Joseph Ishac, Michael Kerrisk, Kostas Pentikousis, Juergen Quittek, Stefan Schmid, Simon Schuetz, and Martin Stiemerling.

10. References

10.1 Normative References

[1] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

[3] Braden, R., "Requirements for Internet Hosts - Communication

Layers", STD 3, RFC 1122, October 1989.

[4] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

[8] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

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

Authors' Addresses

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/

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

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

Appendix A. Document Revision History

Revision	Comments
00	Initial version, merges draft-eggert-tcpm-tcp-abort-timeout-option-00 and draft-gont-tcpm-tcp-auto-option-00.

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

Copyright Statement

Acknowledgment

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