TCP Maintenance and Minor
Extensions (tcpm)
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

M. Kuehlewind
University of Stuttgart
R. Scheffenegger, Ed.
NetApp, Inc.
March 7, 2011

Additional negotiation in the TCP Timestamp Option field during the TCP handshake
draft-scheffenegger-tcpm-timestamp-negotiation-00

Abstract

RFC 1323 defines the TSecr field of a SYN packet to be not valid and thus this field will always be zero. This documents specifies the use of this field to signal and negotiate additional information about the content of the TSopt field as well as the behavior of the receiver. If the receiver understands this extension, it will use the TSecr field of the SYN/ACK to reply. Otherwise the receiver will ignore the TSecr field and set a timestamp in the TSecr field as specified in RFC 1323.

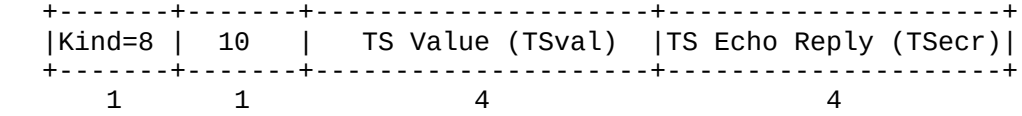
Table of Contents

1. Introduction

The TCP Timestamps Option (TSopt) provides timestamp echoing for Round-trip Time (RTT) measurments. TSopt is widely deployed and activated by default in many systems. RFC 1323 [RFC1323] specifies TSopt the following way:

Kind: 8

Length: 10 bytes



RFC1323 TSopt

"The Timestamps option carries two four-byte timestamp fields. The Timestamp Value field (TSval) contains the current value of the timestamp clock of the TCP sending the option.

The Timestamp Echo Reply field (TSecr) is only valid if the ACK bit is set in the TCP header; if it is valid, it echos a timestamp value that was sent by the remote TCP in the TSval field of a Timestamps option. When TSecr is not valid, its value must be zero. The TSecr value will generally be from the most recent Timestamp option that was received; however, there are exceptions that are explained below.

A TCP may send the Timestamps option (TSopt) in an initial SYN segment (i.e., segment containing a SYN bit and no ACK bit), and may send a TSopt in other segments only if it received a TSopt in the initial SYN segment for the connection."

The comparison of the timestamp in the TSecr field to the current time gives an estimation of the RTT. RFC 1323 [RFC1323] specifies various cases when more than one timestamp is available to echo. The proposed solution might not always be the best choice, e.g. when the TCP Selective Acknowledgment Option (SACK) is used. Moreover, more and more use cases arise where one-way delay (OWD) measurements are needed. These mechanism misuse usually the TSopt to estimated the variation in OWD. To enable such mechanisms the TSecr field in the TCP SYN packet could be used for additional negotiation.

1.1. Requirements Language

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 [RFC2119].

2. Overview

3. Definitions

The reader is expected to be familiar with the definitions given in [RFC1323].

4. Signaling

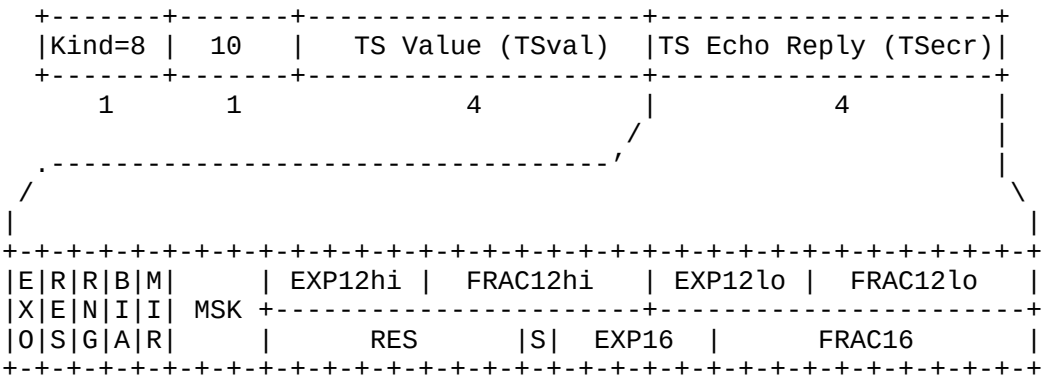
During the initial TCP three-way handshake, timestamp options are negotiated using the TSecr field. A compliant TCP receiver will XOR the flags with the received TSval, when responding with the SYN+ACK. Timestamp Options MAY only be present when the SYN bit is set.

4.1. Capability Flags

In order to signal the supported capabilities, the TSecr is overloaded with the following flags and fields during the three-way handshake. If optional capabilities such as tcp clock range are presented, minimal state will be required in the host to decode the returned Flags xor'ed with the TSval.

Kind: 8

Length: 10 bytes



timestamp option flags

EXO - Extended Options
Indicated that the sender supports extended timestamp options as defined by this document, and MUST be set ("1") by compliant implementations.

RES - Reserved
Reserved for future use. MUST not be set ("0"). If a timestamp option is received with this bit set, the receiver MUST ignore the extended options field and react as if the Flags were not set (compatibility mode).

RNG - Range negotiation
Indicated that the sender is capable of adjusting the timestamp clock rate within the bounds of the two 12 bit fields (see). Only the active sender of a TCP session is allowed to offer a range, while the receiver MAY choose a rate within these bounds.

BIA - Exponent Bias
When set, the 16 and 12 bit floating point exponents are presented with a bias of 21 instead of 15. This allows negotiation of extremely fine-grained timestamp clock resolutions, for example in hardware implementations and high speed (>10 Gigabit/s) environments. See section for more details.

MIR - Always Mirror Timestamp
To disambiguate segements and aid timing calculations even during loss episodes, the timestamp will always be mirrored regardless of the state of the receiver. A sender SHOULD use this option only in conjunction with Selective Acknowledgements (SACK [RFC2018]).

MSK - Mask Timestamps
If the timestamp is used for congestion control purposes, an incentive exists for malicious receivers to reflect tampered timestamps. A sender MAY choose to protect timestamps from such modifications by including a fingerprint (secure hash of some kind) in some of the least significant bits. However, doing so would prevent a receiver from using the timestamp for other purposes. The MASK field indicates how many least significant nibbles should be excluded by the receiver, when processing a timestamp. Note that this does not impact the reflected timestamp in any way - TSecr will always be equal to a appropriate TSval. Another use case would be when the sender does not support a timestamp clock which can guarantee unique timestamps for retransmitted segments. For unambigously identifying regular from retransmitted segments, the timestamp must be unique for otherwise identical segments. Reserving the

lowest nibble for this purpose allows senders with slow running timestamp clocks to make use of this feature.

S - binary16 Sign
This is the sign bit of the IEEE 754-2008 binary16 floating point representation of the timestamp clock. Timestamp clocks MUST be positive, thus this bit MUST be zero.

EXP16 - binary16 Exponent
The exponent component of a binary16 floating point number indicating the timestamp clock. When BIA is not set, the exponent bias is 15 (identical to the binary16 definition in IEEE 754-2008). If OFF is set, the exponent bias is 21, allowing faster timestamp clock rates. Subnormal numbers (lower precision), where the exponent is zero, extend the range to 2^-24 and 2^-30 respectively. Infinity and NaN (all exponent bits set) MUST NOT be invalid, and a timestamp option with NaN/Infinity SHOULD be ignored.

FRAC16 - binary16 Fraction
The fraction component of a binary16 floating point number indicating the timestamp clock. The clock rate is measured in seconds between ticks. The least significant bit corresponds therefore to a time interval of 59.6 ns with the default bias of 15, and 0.931 ns with bias set to 21. The longest time interval would be 65504 sec with default bias, and 511.75 sec with bias set to 21.

EXP12hi and

EXP12lo - binary12 Exponent
The exponent component of a truncated, 12 bit floating point number indicating the possible timestamp clock ranges. Only the host initiating a TCP session MAY offer a timestamp clock range, while the receiver SHOULD select a timestamp clock within these bounds. If the receiver can not adjust it’s timestamp clock to match the range, it MAY use a timestamp clock rate outside these bounds. If the receiver indicated a timestamp clock rate within the indicated bounds, the sender MUST set it’s timestamp clock rate to the negotiated rate. If the receiver uses a timestamp clock rate outside the indicated bounds, it MUST NOT use timestamps where knowledge of the timestamp clock rate is required (ie. congesion control). The exponent bias is 15 when BIA is not set, and 21 otherwise.

FRAC12hi and

FRAC12lo - binary12 Fraction
The fraction component of a 12 bit floating point number. Subnormal numbers are allowed, while Inifinity/NaN MUST NOT be used. Timestamp options with Infinity/NaN values SHOULD be ignored. The smallest representable value is 238 ns with default bias, and 3.73 ns with bias set to 21, while the largest values would be virtually identical to the 16 bit floating point values (65024 and 508 sec).

5. Discussion

One-way delay (variation) based congestion controls would benefit from knowing the clock resolution on both sides.

RTT variance during loss episodes is not deeply researched. Current heuristics (RFC1122, RFC1323, Karn's algorithm, RFC2988) explicitly exclude (and prevent) the use of RTT samples when loss occurs. However, solving the retransmission ambiguity problem - and the related reliable ACK delivery problem - may allow the refinement of these algorithms further, as well as enabling new research to distinguish between corruption loss (without RTT / one-way delay impact) and congestion loss (with RTT / one-way delay impact). Research into this field appears to be a rather neglected, especially when it comes to large scale, public internet investigations. Due to the very nature of this, passive investigations without signals contained within the headers are only of limited use in empirical research.

Retransmission ambiguity detection during loss recovery would allow an additional level of loss recovery control without reverting to timer-based methods. As with the deployment of SACK, separating "what" to send from "when" to send it could be driven one step further. In particular, less conservative loss recovery schemes which do not trade principles of packet conservation against timeliness, require a reliable way of prompt and best possible feedback from the receiver about any delivered segment and their ordering. SACK alone goes quite a long way, but using Timestamp information in addition could remove any ambiguity. However, the current specs in RFC1323 make that use impossible, thus a modified signaling (receiver behavior) is a necessity.

6. Acknowledgements

The authors would like to thank Dragana Damjanovic for some initial

thoughts around Timestamps and their extended potential use.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

The algorithm presented in this paper shares security considerations with [RFC1323].

Some implementations address the vulerabilities of [RFC1323], by dedicating a few low-order bits of the timestamp fields for use with a (secure) hash, that protects against malicious tweaking of TSecr values. A Flag-field has been provided to transparently notify the receiver about that use of low-order bits, so that they can be excluded in one-way delay calculations.

9. References

9.1. Normative References

[RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[Chirp] Kuehlewind, M. and B. Briscoe, "Chirping for Congestion Control - Implementation Feasibility", Nov 2010, <http://bobbriscoe.net/projects/netsvc_i-f/chirp_pfldnet10.pdf>.

[I-D.ietf-tcpm-tcp-security] Gont, F., "Security Assessment of the Transmission Control Protocol (TCP)", draft-ietf-tcpm-tcp-security-02 (work in progress), January 2011.

Authors' Addresses

Mirja Kuehlewind
University of Stuttgart
Pfaffenwaldring 47
Stuttgart 70569
Germany

Email: mirja.kuehlewind@ikr.uni-stuttgart.de

Richard Scheffenegger (editor)
NetApp, Inc.
Am Euro Platz 2
Vienna, 1120
Austria

Phone: +43 1 3676811 3146
Email: rs@netapp.com