Multipath TCP timestamp option
draft-bonaventure-mptcp-timestamp-01

Abstract

The TCP timestamps defined in [RFC1323] were designed to improve round-trip-time estimations and provide protection against wrapped sequence numbers (PAWS). This draft clarifies the utilisation of timestamps by Multipath TCP and proposes a new timestamp option that better suits the needs of Multipath TCP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 3, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

The Timestamps option was proposed in [RFC1323]. Each Timestamps option contains two timestamps. The first one corresponds to the current value of the sender’s clock. The second timestamp allows to echo the most recent timestamp received from the remote host. The utilisation of this option can be negotiated on a per-connection basis during the three-way handshake. The timestamps option was motivated by two usages:

- improve the accuracy of the round-trip-time measurements
- provide protection against wrapped TCP sequence numbers (PAWS)

Although these two usages have completely different purposes, they are coupled in [RFC1323]. [RFC7323] goes further by requiring that the TCP timestamps option be included in all segments once the option has been negotiated in the three-way handshake. Forcing the utilisation of this option in all segments is required to support PAWS. However, there is no reason to force TCP hosts to include the timestamp option in all segments when PAWS is not required.

In practice, there are two important use cases where PAWS is not required. The first is when the TCP connections are so short that TCP sequence numbers cannot wrap around. The second use case is when Multipath TCP is used. Multipath TCP, defined in [RFC6824], is a TCP extension that enables a TCP connection to exchange data over multiple paths. This TCP extension uses 64 bits sequence number which solves the PAWS problem in a cleaner way than [RFC7323]. Once Multipath TCP has been negotiated, the PAWS part of [RFC1323] becomes useless and should be disabled.

8.1. Normative References

8.2. Informative References
This document is organised as follows. We first summarize in section Section 2 the issues with the TCP timestamps option defined in [RFC1323]. We then propose in section Section 3 a Multipath TCP Timestamp option that should be used by Multipath TCP implementations instead of the regular [RFC7323] timestamps options.

2. The TCP Timestamps option and Multipath TCP

The TCP timestamps option defined in [RFC1323] is encoded as shown in figure Figure 1.

+-------+-------+---------------------+---------------------+
|Kind=8 | 10    |   TS Value (TSval)  |TS Echo Reply (TSecr) |
+-------+-------+---------------------+---------------------+

Figure 1: Original RFC1323 Timestamps option

This option consumes 10 bytes. When [RFC1323] was published, consuming 10 bytes in the SYN segment to negotiate the utilisation of this option and later in all data segments was not a severe concern given the limited number of TCP options that were used at that time. This is not anymore the case today with Multipath TCP [RFC6824] or the Selective Acknowledgements [RFC2018] option.

A Multipath TCP implementation SHOULD not use the [RFC7323] timestamps option on Multipath TCP connections. However, a regular TCP connection SHOULD use the [RFC7323] timestamp option to protect against wrapped sequence numbers. To achieve this objective, Multipath TCP implementations SHOULD operate as follows:

- an active Multipath TCP opener SHOULD place both the Timestamps and MP_CAPABLE options in SYN segments when trying to open a TCP connection unless the remote host (and the path towards this host) is known to support Multipath TCP. In this case, the Timestamps option can be ignored in the SYN segment.

- a passive Multipath TCP opener that receives a SYN segment containing both the Timestamp and the MP_CAPABLE options SHOULD only include the MP_CAPABLE option in the returned SYN+ACK segment. This would disable the [RFC7323] timestamps on the Multipath TCP connection.

When creating subflows, the Timestamps option SHOULD NOT be associated with the MP_JOIN option in the SYN segments. Furthermore, if a Multipath TCP host receives a valid SYN segment that contains...
both the MP_JOIN option and the Timestamps option, it should not include the Timestamps option in the returned SYN+ACK segment.

3. The Multipath TCP Timestamp option

The Timestamps option defined in [RFC7323] encodes two 32 bits timestamps. Having two timestamps is useful when data transfer is bidirectional but in practice very few TCP connections are totally bidirectional. Most TCP connections send data in one directions and acknowledgments in the opposite. For these connections, placing two timestamps in each segment that carries data and each acknowledgement is a waste of TCP options space.

When precise round-trip-time measurements are required on a Multipath TCP connection, those measurements can be performed by using the experimental Multipath TCP option shown in figure Figure 2.

```
<table>
<thead>
<tr>
<th>Kind</th>
<th>Length</th>
<th>Subtype</th>
<th>Flags</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id. (16 bits)</td>
<td>Timestamp (24 bits)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 2: The experimental MPTCP Timestamp Option

The experimental MPTCP TS option contains the flags defined in [I-D.bonaventure-mptcp-exp-option].

The experimental MPTCP TS option may be sent in any TCP segment except those having the SYN flag set.

When a host receives a segment containing an MPTCP TS option whose ‘S’ flag is set, it SHOULD reply immediately by echoing the received timestamp in a returned MPTCP TS option whose ‘S’ flag is reset. This MPTCP TS option can be included in either a segment that carries data, if one is pending, or an acknowledgement. This implies that a host does not need to maintain additional state to process the received MPTCP TS option since it can reply directly to any received MPTCP TS option.

The MPTCP TS option can be used to improve the quality of the round-trip-time estimator. The discussion in section 4.2 of [RFC7323] is also applicable for the Timestamp proposed in this document.
The MPTCP TS may also be used to verify that a subflow remains active by forcing a remote host to reply to an MPTCP segment without sending data.

4. Security Considerations

A middlebox may remove the experimental MPTCP TS option. This is unlikely if the Multipath TCP connection operates correctly. Since the MPTCP TS option is only informational, such a behaviour would not affect the reliability of the Multipath TCP connection.

Some of the security considerations from [RFC7323] and in particular the following paragraph apply for the MPTCP TS option:

A naive implementation that derives the timestamp clock value directly from a system uptime clock may unintentionally leak this information to an attacker. This does not directly compromise any of the mechanisms described in this document. However, this may be valuable information to a potential attacker. It is therefore RECOMMENDED to generate a random, per-Multipath TCP connection offset to be used with the clock source when generating the Timestamps option value.

By carefully choosing this random offset, further improvements as described in [RFC6191] are possible.

5. IANA Considerations

This document proposes an experimental MPTCP option to carry timestamps. If [I-D.bonaventure-mptcp-exp-option] is approved, then an experimental identifier should be added to the IANA registry to identify the timestamp option.

6. Conclusion

This document has proposed a new Timestamp option to replace the [RFC7323] Timestamps option on Multipath TCP connections. The MPTCP TS option can be included in MPTCP segments only when needed to preserve TCP option space notably for the MPTCP and SACK options.

7. Acknowledgements

This work was partially supported by the EC within the FP7-Trilogy2 project. The author would like to thank Raphael Bauduin for his comments and Joe Touch whose comments on the mptcp mailing list encouraged the writing of this draft.
8. References

8.1. Normative References

[I-D.bonaventure-mptcp-exp-option]
Bonaventure, O., benjamin.hesmans@uclouvain.be, b., and M.
Boucadair, "Experimental Multipath TCP option", draft-
bonaventure-mptcp-exp-option-00 (work in progress), June
2015.


[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure,
"TCP Extensions for Multipath Operation with Multiple

Scheffenegger, "TCP Extensions for High Performance", RFC
7323, September 2014.

8.2. Informative References


[RFC6191] Gont, F., "Reducing the TIME-WAIT State Using TCP
Timestamps", BCP 159, RFC 6191, April 2011.

Appendix A. Changelog

This appendix should be removed before publication.

Changes in version 01.

- updated the format of the proposed option to use the encoding
  proposed in [I-D.bonaventure-mptcp-exp-option]

Author’s Address

Olivier Bonaventure
UCLouvain

Email: Olivier.Bonaventure@uclouvain.be