                         NTP Interleaved Modes
                 draft-mlichvar-ntp-interleaved-modes-01

Abstract

   This document extends the specification of Network Time Protocol
   (NTP) version 4 in RFC 5905 with special modes called the NTP
   interleaved modes, that enable NTP servers to provide their clients
   and peers with more accurate transmit timestamps that are available
   only after transmitting NTP packets. More specifically, this
   document describes three modes: interleaved client/server,
   interleaved symmetric, and interleaved broadcast.

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 https://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 June 15, 2018.

Copyright Notice

1. Introduction

   RFC 5905 [RFC5905] describes the operations of NTPv4 in basic client/
   server, symmetric, and broadcast mode. The transmit timestamp is one
   of the four timestamps included in every NTP packet used for time
   synchronization. A packet that strictly follows RFC 5905, i.e. it
   contains a transmit timestamp corresponding to the packet itself, is
   said to be in basic mode.

   There are, at least, four options where a transmit timestamp can be
   captured i.e. by NTP daemon, by network drivers, or at the MAC or
   physical layer of the OSI model. A typical transmit timestamp in a
   software NTP implementation in the basic mode is the one captured by
   the NTP daemon using the system clock, before the computation of
   message digest and before the packet is passed to the operating
   system, and does not include any processing and queuing delays in the
   system, network drivers, and hardware. These delays may add a
   significant error to the offset and network delay measured by clients
   and peers of the server.

   For best accuracy, the transmit timestamp should be captured as close
   to the wire as possible, but that is difficult to implement in the
   current packet since this timestamp is available only after the
   packet transmission. The protocol described in RFC 5905 does not
   specify any mechanism for the server to provide its clients and peers
   with this more accurate timestamp.

   Different mechanisms could be used to exchange this more accurate
   timestamp. This document describes interleaved modes, in which an
   NTP packet contains a transmit timestamp corresponding to the
   previous packet that was sent to the client or peer. This transmit
   timestamp could be captured at one of the any four places mentioned
   above. More specifically, this document:

   1. Introduces and specifies a new interleaved client/server mode.

   2. Specifies the interleaved symmetric mode based on the NTP
       reference implementation with some modifications.

   3. Specifies the interleaved broadcast mode based purely on the NTP
       reference implementation.

   The protocol does not change the NTP packet header format. Only the
   semantics of some timestamp fields is different. NTPv4 that supports

client/server and broadcast interleaved modes is compatible with
NTPv4 without this capability as well as with all previous NTP
versions.

The protocol requires both servers and clients/peers to keep some
state specific to the interleaved mode. It prevents traffic
amplification that would be possible if the timestamp was sent in a
separate message in order to keep the servers stateless.

This document assumes familiarity with RFC 5905.

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 RFC 2119 [RFC2119].

2. Interleaved Client/server mode

The interleaved client/server mode is similar to the basic client/
server mode. The only difference between the two modes is in the
meaning of the transmit and origin timestamp fields.

A client request in the basic mode has an origin timestamp equal to
the transmit timestamp from the previous server response, or is zero.
A server response in the basic mode has an origin timestamp equal to
the transmit timestamp from the client's request. The transmit
timestamps correspond to the packets in which they are included.

A client request in the interleaved mode has an origin timestamp
equal to the receive timestamp from the previous server response. A
server response in the interleaved mode has an origin timestamp equal
to the receive timestamp from the client's request. The transmit
timestamps correspond to the previous packets that were sent to the
server or client.

A server which supports the interleaved mode needs to save pairs of
local receive and transmit timestamps. The server SHOULD discard old
timestamps to limit the amount of memory needed to support clients
using the interleaved mode. The server MAY separate the timestamps
by IP addresses, but it SHOULD NOT separate them by port numbers,
i.e. clients are allowed to change their source port between
requests.

When the server receives a request, it SHOULD compare the origin
timestamp with all receive timestamps it has saved (for the IP
address). If a match is found, the server SHOULD respond with a
packet in the interleaved mode, which contains the transmit timestamp

corresponding to the packet which had the matching receive timestamp. If no match is found, the server MUST NOT respond in the interleaved mode. The server MAY always respond in the basic mode. In both cases, the server SHOULD save the new receive and transmit timestamps.

Both servers and clients that support the interleaved mode MUST NOT send a packet that has a transmit timestamp equal to the receive timestamp in order to reliably detect whether received packets conform to the interleaved mode.

The first request from a client is always in the basic mode and so is the server response. It has a zero origin timestamp and zero receive timestamp. Only when the client receives a valid response from the server, it will be able to send a request in the interleaved mode. The client SHOULD limit the number of requests in the interleaved mode per server response to prevent processing of very old timestamps in case a large number of packets is lost.

An example of packets in a client/server exchange using the interleaved mode is shown in Figure 1. The packets in the basic and interleaved mode are indicated with B and I respectively. The timestamps t1', t3' and t11' point to the same transmissions as t1, t3 and t11, but they may be less accurate. The first exchange is in the basic mode followed by a second exchange in the interleaved mode. For the third exchange, the client request is in the interleaved mode, but the server response is in the basic mode, because the server did not have the pair of timestamps t6 and t7 (e.g. they were dropped to save timestamps for other clients using the interleaved mode).

```
Server   t2   t3                 t6   t7                 t10  t11
    -----+----+----------------+----+----------------+----+-----
         /     \               /     \               /     \
Client /        \             /       \             /       \
    --+----------+----------+----------+----------+----------+--
      t1         t4         t5         t8         t9         t12

Mode: B          B          I          I          I          B
    +----+     +----+     +----+     +----+     +----+     +----+
Org | 0  |     | t1'|     | t2 |     | t4 |     | t6 |     | t5 |
Rx  | 0  |     | t2 |     | t4 |     | t6 |     | t8 |     |t10 |
Tx  | t1'|     | t3'|     | t1 |     | t3 |     | t5 |     |t11'|
    +----+     +----+     +----+     +----+     +----+     +----+
```
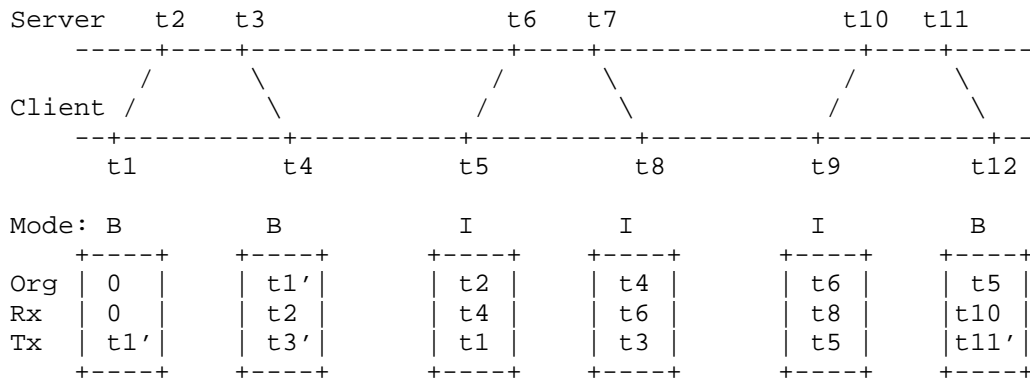
Figure 1: Packet timestamps in interleaved client/server mode

When the client receives a response, it performs all tests described
in RFC 5905, except now the sanity check for bogus packet needs to
compare the origin timestamp with both transmit and receive
timestamps from the request in order to be able to detect if the
response is in the basic or interleaved mode. The client SHOULD NOT
update its NTP state when an invalid response is received to not lose
the timestamps which will be needed to complete a measurement when
the following response in the interleaved mode is received.

If the packet passed the tests and conforms to the interleaved mode,
the client can compute the offset and delay using the formulas from
RFC 5905 and one of two different sets of timestamps. The first set
is RECOMMENDED for clients that filter measurements based on the
delay. The corresponding timestamps from Figure 1 are written in
parentheses.

    T1 - local transmit timestamp of the previous request (t1)

    T2 - remote receive timestamp from the previous response (t2)

    T3 - remote transmit timestamp from the latest response (t3)

    T4 - local receive timestamp of the previous response (t4)

The second set gives a more accurate measurement of the current
offset, but the delay is much more sensitive to a frequency error
between the server and client due to a much longer interval between
T1 and T4.

    T1 - local transmit timestamp of the latest request (t5)

    T2 - remote receive timestamp from the latest response (t6)

    T3 - remote transmit timestamp from the latest response (t3)

    T4 - local receive timestamp of the previous response (t4)

Clients MAY filter measurements based on the mode. The maximum
number of dropped measurements in the basic mode SHOULD be limited in
case the server does not support or is not able to respond in the
interleaved mode. Clients that filter measurements based on the
delay will implicitly prefer measurements in the interleaved mode
over the basic mode, because they have a shorter delay due to a more
accurate transmit timestamp (T3).

The server MAY limit saving of the receive and transmit timestamps to
requests which have an origin timestamp specific to the interleaved
mode in order to not waste resources on clients using the basic mode.

Such an optimization will delay the first interleaved response of the
server to a client by one exchange.

A check for a non-zero origin timestamp works with clients that
implement NTP data minimization [I-D.ietf-ntp-data-minimization]. To
detect requests in the basic mode from clients that do not implement
the data minimization, the server can encode in low-order bits of the
receive and transmit timestamps below precision of the clock a bit
indicating whether the timestamp is a receive timestamp. If the
server receives a request with a non-zero origin timestamp which does
not indicate it is receive timestamp of the server, the request is in
the basic mode and it is not necessary to save the new receive and
transmit timestamp.

3. Interleaved Symmetric mode

The interleaved symmetric mode uses the same principles as the
interleaved client/server mode. A packet in the interleaved
symmetric mode has a transmit timestamp which corresponds to the
previous packet sent to the peer and an origin timestamp equal to the
receive timestamp from the last packet received from the peer.

In order to prevent the peer from matching the transmit timestamp
with an incorrect packet when the peers' transmissions do not
alternate (e.g. they use different polling intervals) and a previous
packet was lost, the use of the interleaved mode in symmetric
associations requires additional restrictions.

Peers which have an association need to count valid packets received
between their transmissions to determine in which mode a packet
should be formed. A valid packet in this context is a packet which
passed all NTP tests for duplicate, replayed, bogus, and
unauthenticated packets. Other received packets may update the NTP
state to allow the (re)initialization of the association, but they do
not change the selection of the mode.

A peer A SHOULD send a peer B a packet in the interleaved mode only
when the following conditions are met:

1. The peer A has an active association with the peer B which was
    specified with an option enabling the interleaved mode, OR the
    peer A received at least one valid packet in the interleaved mode
    from the peer B.

2. The peer A did not send a packet to the peer B since it received
    the last valid packet from the peer B.

3. The previous packet that the peer A sent to the peer B was the
    only response to a packet received from the peer B.

An example of packets exchanged in a symmetric association is shown
in Figure 2. The minimum polling interval of the peer A is twice as
long as the maximum polling interval of the peer B. The first
packets sent by the peers are in the basic mode. The second and
third packet sent by the peer A is in the interleaved mode. The
second packet sent by the peer B is in the interleaved mode, but the
following packets sent by the peer are in the basic mode, because
multiple responses are sent per request.

```
Peer A   t2 t3       t6            t8 t9      t12         t14 t15
    -----+--+--------+-----------+--+--------+-----------+--+-----
         /   \      /            /   \      /            /   \
Peer B  /     \    /            /     \    /            /     \
    --+--------+--+-----------+--------+--+-----------+--------+--
      t1          t4 t5       t7          t10 t11      t13        t16

Mode: B        B        I         B        I        B         B        I
      +----+ +----+ +----+    +----+ +----+ +----+    +----+ +----+
Org | 0    | | t1'| | t2 |    | t3'| | t4 | | t3 |    | t3 | |t10 |
Rx  | 0    | | t2 | | t4 |    | t4 | | t8 | |t10 |    |t10 | |t14 |
Tx  | t1'  | | t3'| | t1 |    | t7'| | t3 | |t11'|    |t13'| | t9 |
      +----+ +----+ +----+    +----+ +----+ +----+    +----+ +----+
```
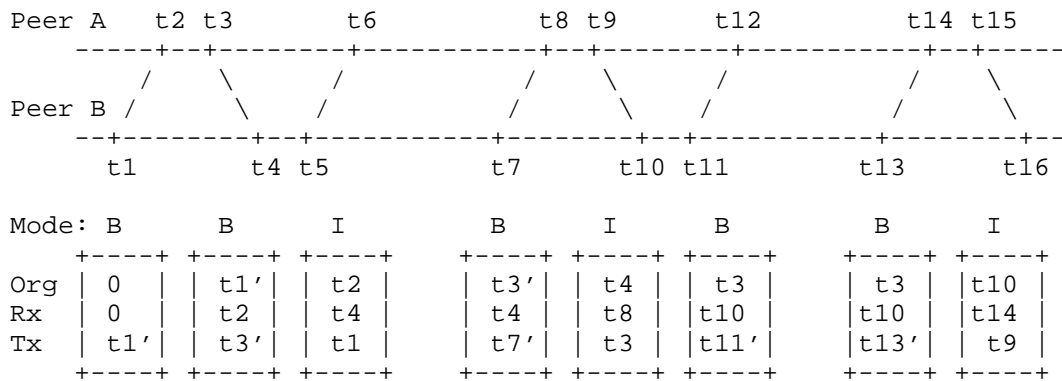
             Figure 2: Packet timestamps in interleaved symmetric mode

If the peer A has no association with the peer B and it responds with
symmetric passive packets, it does not need to count the packets in
order to meet the restrictions, because each request has at most one
response. The peer SHOULD process the requests in the same way as a
server which supports the interleaved client/server mode. It MUST
NOT respond in the interleaved mode if the request was not in the
interleaved mode.

The peers SHOULD compute the offset and delay using one the two sets
of timestamps specified in the client/server section. They MAY
switch between them to minimize the interval between T1 and T4 in
order to reduce the error in the measured delay.

4. Interleaved Broadcast mode

A packet in the interleaved broadcast mode contains two transmit
timestamps. One corresponds to the packet itself and is saved in the
transmit timestamp field. The other corresponds to the previous
packet and is saved in the origin timestamp field. The packet is
compatible with the basic mode, which uses a zero origin timestamp.

A client which does not support the interleaved mode ignores the
origin timestamp and processes all packets as if they were in the
basic mode.

A client which supports the interleaved mode SHOULD check if the
origin timestamp is not zero to detect packets in the interleaved
mode. The client SHOULD also compare the origin timestamp with the
transmit timestamp from the previous packet to detect lost packets.
If the difference is larger than a specified maximum (e.g. 1 second),
the packet SHOULD NOT be used for synchronization.

The client SHOULD compute the offset using the origin timestamp from
the received packet and the local receive timestamp of the previous
packet. If the client needs to measure the network delay, it SHOULD
use the interleaved client/server mode.

5. Acknowledgements

The interleaved modes described in this document are based on the
reference NTP implementation written by David Mills.

The authors would like to thank Kristof Teichel for his useful
comments.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

Security issues that apply to the basic modes apply also to the
interleaved modes. They are described in The Security of NTP's
Datagram Protocol [SECNTP].

Clients and peers SHOULD NOT leak the receive timestamp in packets
sent to other peers or clients (e.g. as a reference timestamp) to
prevent off-path attackers from easily getting the origin timestamp
needed to make a valid response in the interleaved mode.

Clients SHOULD randomize all bits of both receive and transmit
timestamps, as recommended for the transmit timestamp in the NTP
client data minimization [I-D.ietf-ntp-data-minimization], to make it
more difficult for off-path attackers to guess the origin timestamp.

Protecting symmetric associations in the interleaved mode against
replay attacks is even more difficult than in the basic mode, because
the NTP state needs to be protected not only between the reception
and transmission in order to send the peer a packet with a valid

origin timestamp, but all the time to not lose the timestamps which
will be needed to complete a measurement when the following packet in
the interleaved mode is received.

8. References

8.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
              "Network Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
              <https://www.rfc-editor.org/info/rfc5905>.

8.2. Informative References

   [I-D.ietf-ntp-data-minimization]
              Franke, D. and A. Malhotra, "NTP Client Data
              Minimization", draft-ietf-ntp-data-minimization-01 (work
              in progress), July 2017.

   [SECNTP]   Malhotra, A., Gundy, M., Varia, M., Kennedy, H., Gardner,
              J., and S. Goldberg, "The Security of NTP's Datagram
              Protocol", 2016, <http://eprint.iacr.org/2016/1006>.

Authors' Addresses

   Miroslav Lichvar
   Red Hat
   Purkynova 115
   Brno 612 00
   Czech Republic

   Email: mlichvar@redhat.com

   Aanchal Malhotra
   Boston University
   111 Cummington St
   Boston 02215
   USA

   Email: aanchal4@bu.edu