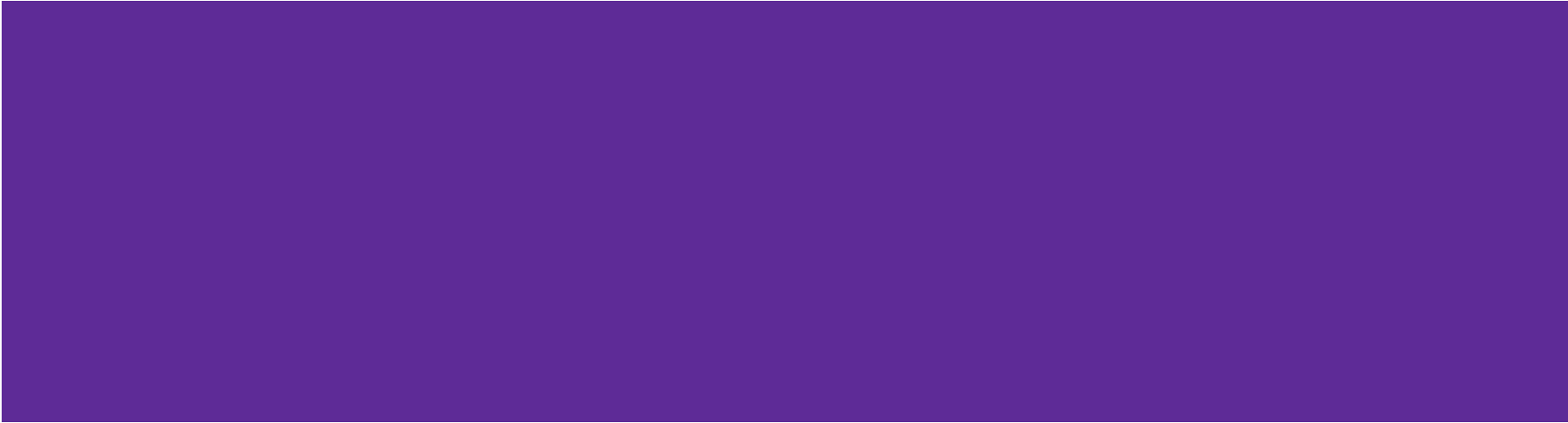


TCP ETS: Extensible Timestamp Option

[draft-yang-tcpm-ets-00](#)

Kevin Yang, Neal Cardwell, Yuchung Cheng, Eric Dumazet
{yyd, ncardwell, ycheng, edumazet}@google.com
tcpm IETF 109 2020-11-17



ETS Design

1. Design to subsume RFC7323
2. Exchange maximum ACK delay (MaxACKDel) in handshake
3. Use finer clock granularity: 1 microsecond per timestamp tick
4. Introduce echo reply delay (EcrDel), for network RTT measurements
5. Allow using NIC HW timestamps for better network RTT measurements

Motivation

(1) Congestion control:

Precise network/host delay measurements: accurate and precise (usec-granularity) delay measurements of network and host can enable simple, effective datacenter congestion control algorithms, especially aided by NIC HW timestamping. Example: Swift [[SIGCOMM20](#)]

(2) Loss recovery:

Legacy 200ms delayed ACKs cause TCP stacks to use a minimum RTO of 200ms.

(3) Pacing:

Pacing rates computed as $(rate = k * cwnd / srtt)$ are slowed (up to 100x) by delayed ACKs.

Problems with RFC7323

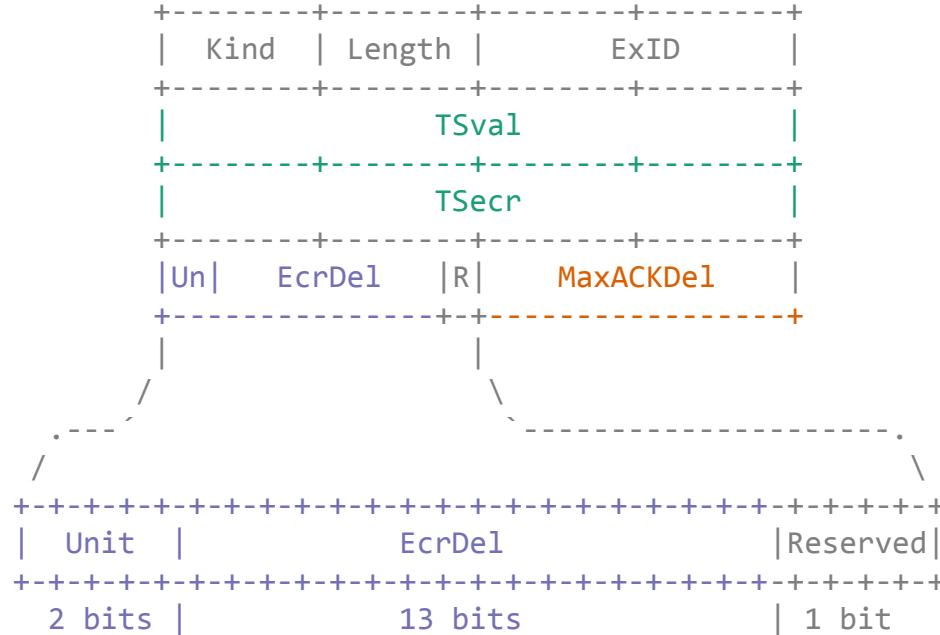
Problems with the [RFC7323](#) Timestamp option:

1. Delays from delayed ACK are included in RTT
2. ACKs can only be used if they advance SND.UNA (the left edge of the send window)[1]
3. Host-side delays are included. For example, delays for waking CPUs from power management "C-states" are included in RTT (can be longer than network RTT)
4. Clock frequency (1 ms to 1 sec) too coarse v.s. modern datacenter networks with RTT < 100us [2] (limiting RTT precision and Eifel spurious retransmit detection)

[1] RFC7323 Section 4.1

[2] Barroso, L., Marty, M., Patterson, D., and P. Ranganathan, "Attack of the Killer Microseconds", Communications of the ACM , April 2017.

ETS TCP Option (ETSOpt) Header Format



Kind: 1 byte, value 254, [RFC6994] experimental option
 Length: 1 byte option length, value is 16 if SYN bit is set, otherwise 14 (value MAY be higher in later versions).

ExID: 2 bytes, [RFC6994] experiment ID: value 0x4554.

TSval and TSecr: 32 bits each, have the same definition as [RFC7323] but are in **microseconds**.

EcrDelUnit: 2 bits; allowed values are:

- 0: indicates EcrDel is in microsecond units
- 1: indicates EcrDel is in millisecond units
- 2: indicates EcrDel is invalid (should be ignored)
- 3: reserved in this protocol version

EcrDel: 13 bits, the value of EcrDel.

Reserved: 1 bit, in this protocol version, sender MUST set to 0
 And receiver MUST ignore.

MaxACKDel: 16 bits, max expected ACK delay in microseconds, **only present in SYN**.

Examples: handshake

TCP A (Client)

TCP B (Server)

CLOSED

LISTEN

#1 SYN-SENT

--- <SYN,TSval=X,TSecr=0,
EcrDel=0,MaxACKDel=M1> -----> SYN-RCVD

#2 ESTABLISHED

<SYN,ACK,TSval=Y,TSecr=X, ----- SYN-RCVD
<-- EcrDel=E1,MaxACKDel=M2>

#3 ESTABLISHED

-- <ACK,TSval=Z,TSecr=Y,EcrDel=E2> --> ESTABLISHED

ETSopt: How Endpoints Compute EcrDel

TCP A (Client)

TCP B (Server)

ESTABLISHED -- <ACK,TSval=Z,TSecr=Y,EcrDel=E2> --> ESTABLISHED

TCP endpoints compute EcrDel using the following algorithm:

(1) When a <SYN> or non-empty data segment SEG is received:

 If SYN is set, or SEG.TSval is after TS.Latest:

 TS.Latest = SEG.TSval

 TS.LatestClock = ArrivalTime of SEG

(2) When an ETSopt is sent:

 TSecr = TS.Recent (as in [\[RFC7323\]](#))

 LatestACKDel = ACKSendTime - TS.LatestClock

 TSecrAge = TS.Latest - TSecr

EcrDel = LatestACKDel + TSecrAge

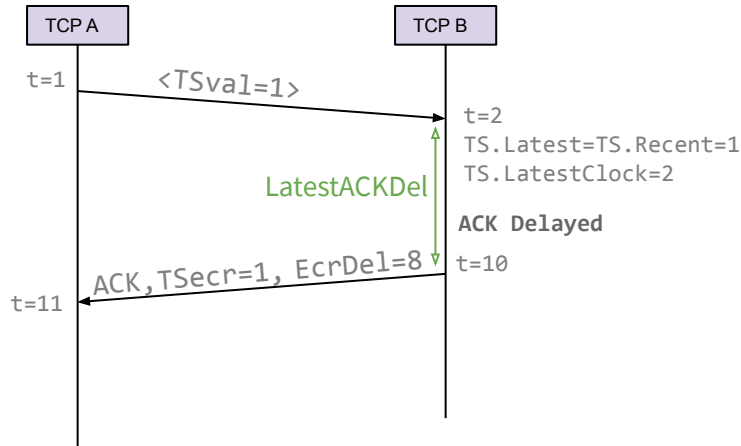
Note: EcrDel is 0 when (1) there is no ACK delay, and (2) TSecr is the latest received timestamp

ETSopt: Network RTT

Network RTT: the time from when the data segment leaves the sender until when it arrives at the receiver, plus the time from when the corresponding ACK leaves the (data) receiver until the ACK arrives at the data sender.

Network RTT is estimated by:

$$\text{NetworkRTT} = \text{ACKArrivalTime} - \text{SEG.TSecr} - \text{SEG.EcrDel}$$



$$\text{LatestACKDel} = \text{ACKSendTime} - \text{TS.LatestClock}$$

$$\text{TSecrAge} = \text{TS.Latest} - \text{TSecr}$$

$$\text{EcrDel} = \text{LatestACKDel} + \text{TSecrAge}$$

$$= (10 - 2) + (1 - 1)$$

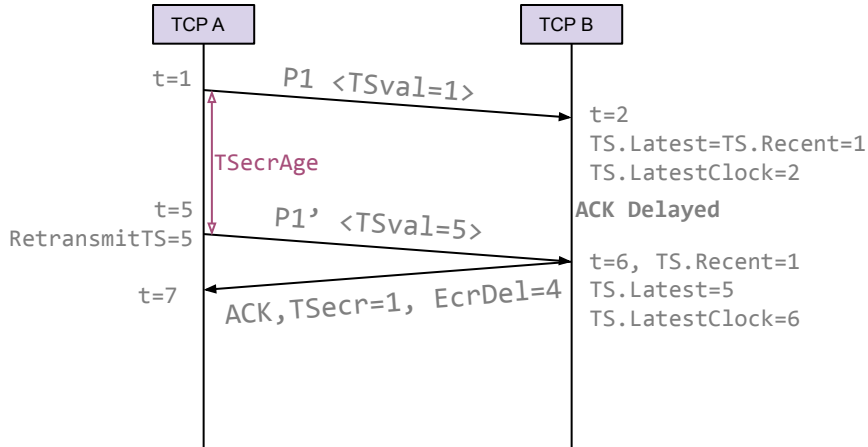
$$= 8$$

$$\text{NetworkRTT} = \text{ACKArrivalTime} - \text{SEG.TSecr} - \text{SEG.EcrDel}$$

$$= 11 - 1 - 8$$

$$= 2$$

Examples: Eifel Detection of Spurious Retx



$$\begin{aligned}
 \text{LatestACKDel} &= \text{ACKSendTime} - \text{TS.LatestClock} \\
 \text{TSecrAge} &= \text{TS.Latest} - \text{TSecr} \\
 \text{EcrDel} &= \text{LatestACKDel} + \text{TSecrAge} \\
 &= (6 - 6) + (5 - 1) \\
 &= 4
 \end{aligned}$$

(1) Sender can detect spurious retransmit:

$\text{TSecr} < \text{RetransmitTS}$ ($1 < 5$), thus the sender concludes the retransmission of `P1'` is spurious, via Eifel [\[RFC3522\]](#)

(2) Sender can compute accurate network RTT:

$$\begin{aligned}
 \text{NetworkRTT} &= \text{ACKArrivalTime} - \text{SEG.TSecr} - \text{SEG.EcrDel} \\
 &= 7 - 1 - 4 \\
 &= 2, \text{ is the RTT of } P1'
 \end{aligned}$$

Exchanging the Maximum ACK delay

When an ACK is overdue, a sender does not know whether...

- The packet was lost, or

- The ACK is being delayed by the receiver

A sender needs to “guess” the maximum ACK delay of remote receiver
to avoid spurious retransmits and spurious congestion control reactions

Legacy 200ms[1] delayed ACKs cause TCP stacks to use a minimum RTO of 200ms.

In ETS, the maximum ACK delay (MaxACKDel) is exchanged in handshakes.

- This allows a much smaller minimum RTO (e.g. 5ms)

[1] Wright, G. and W. Stevens, "TCP/IP Illustrated, Volume 2: The Implementation", 1995.

Discussion (part 1)

Protection Against Wrapped Sequences (PAWS) check: skip check if idle for ≥ 2147 sec

Eifel: unaffected: Eifel works the same with ETS as RFC7323 TS

Retransmission timeout calculation: unaffected: update srtt using $RTT = now - TSEcr$

Space left for SACK blocks: ETS leaves space for 3 SACK blocks (just like RFC7323)

Discussion (part 2)

Middlebox/compatibility considerations:

- SYN MAY include both ETS and RFC7323 TS...
 - In case ETS option is stripped
 - In case receiver only understands RFC7323 TS
- SYN and SYN+ACK retransmit MAY use just RFC7323 TS, in case ETS SYNs dropped

Security: no new issues

Future extensions: future revisions can include more data using a longer length field

Q&A

