



# TCP Timers

Difficulty Level : Medium • Last Updated : 30 Oct, 2020

TCP uses several timers to ensure that excessive delays are not encountered during communications. Several of these timers are elegant, handling problems that are not immediately obvious at first analysis. Each of the timers used by TCP is examined in the following sections, which reveal its role in ensuring data is properly sent from one connection to another.

## TCP implementation uses four timers –

1. **Retransmission Timer** – To retransmit lost segments, TCP uses retransmission timeout (RTO). When TCP sends a segment the timer starts and stops when the acknowledgment is received. If the timer expires timeout occurs and the segment is retransmitted. RTO (retransmission timeout is for 1 RTT) to calculate retransmission timeout we first need to calculate the RTT(round trip time).

### RTT three types –

- **Measured RTT(RTTm)** – The measured round-trip time for a segment is the time required for the segment to reach the destination and be acknowledged, although the acknowledgment may include other segments.
- **Smoothed RTT(RTTs)** – It is the weighted average of RTTm. RTTm is likely to change and its fluctuation is so high that a single measurement cannot be used to calculate RTO.



Initially -> No value

After the first measurement ->  $RTTs = RTT_m$

After each measurement ->  $RTTs = (1-t) * RTTs + t * RTT_m$

Note:  $t = 1/8$  (default if not given)

- **Deviated RTT ( $RTT_d$ )** – Most implementations do not use RTTs alone so RTT deviated is also calculated to find out RTO.

Initially -> No value

After the first measurement ->  $RTT_d = RTT_m / 2$

After each measurement ->  $RTT_d = (1-k) * RTT_d + k * (RTT_m - RTTs)$

Note:  $k = 1/4$  (default if not given)

**Retransmission Timeout : RTO calculation** – The value of RTO is based on the smoothed round-trip time and its deviation. Most implementations use the following formula to calculate the RTO:

Initial value -> Original (given in question)

After any measurement ->  $RTO = RTTs + 4 * RTT_d$

#NOTE: At every retransmission the value of RTO doubles. ( $RTO(\text{new}) = RTO(\text{before retransmission}) * 2$ ) this is explained in [Karn's Algorithm](#)

2. **Persistent Timer** – To deal with a zero-window-size deadlock situation, TCP uses a persistence timer. When the sending TCP receives an acknowledgment with a window size of zero, it starts a persistence timer. When the persistence timer goes off, the sending TCP sends a special segment called a probe. This segment contains only 1 byte of new data. It has a sequence number, but its acknowledgment number is never acknowledged; it is even ignored



in calculating the sequence number for the rest of the data. The probe causes the receiving TCP to resend the acknowledgment which was lost.

3. **Keep Alive Timer** – A keepalive timer is used to prevent a long idle connection between two TCPs. If a client opens a TCP connection to a server transfers some data and becomes silent the client will crash. In this case, the connection remains open forever. So a keepalive timer is used. Each time the server hears from a client, it resets this timer. The time-out is usually 2 hours. If the server does not hear from the client after 2 hours, it sends a probe segment. If there is no response after 10 probes, each of which is 75 s apart, it assumes that the client is down and terminates the connection.

4. **Time Wait Timer** – This timer is used during [tcp connection termination](#). The timer starts after sending the last Ack for 2nd FIN and closing the connection.

*After a TCP connection is closed, it is possible for datagrams that are still making their way through the network to attempt to access the closed port. The quiet timer is intended to prevent the just-closed port from reopening again quickly and receiving these last datagrams.*

The **quiet timer** is usually set to twice the maximum segment lifetime (the same value as the Time-To-Live field in an IP header), ensuring that all segments still heading for the port have been discarded.

## Reference –

### [TCP Timers – Que10](#)

This article is contributed by [SHAURYA UPPAL](#). If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://www.geeksforgeeks.org/contribute) or mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



**Like** 0

Previous

Next

Related Articles

Save for later

[ADVERTISEMENT BY ADRECOVER](#) [ADVERTISE ON GEEKSFORGEEEKS](#)

## RECOMMENDED ARTICLES



Page : 1 2 3

**01** TCP with explicit link failure notification (TCP-ELFN)  
06, Jan 20

**02** Why does DNS use UDP and not TCP?  
11, Jul 15

**03** Basic concept of TCP-Vegas  
31, Dec 20

**04** Error Control in TCP  
18, Oct 17

**05** TCP/IP Model  
04, Oct 17

**06** TCP 3-Way Handshake Process  
05, Oct 17

**07** TCP Connection Termination  
18, Oct 17

**08** TCP Congestion Control  
31, Oct 17

### Article Contributed By :



### Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [Palak Jain 5](#), [shashankrustagi\(shanu\)](#)

Article Tags : [Computer Networks](#), [GATE CS](#)

Practice Tags : [Computer Networks](#)



[Improve Article](#)[Report Issue](#)

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

[Load Comments](#)

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)[Careers](#)

## Learn

[Algorithms](#)[Data Structures](#)

## Practice

[Courses](#)[Company-wise](#)

## Contribute

[Write an Article](#)[Write Interview Experience](#)

[Privacy Policy](#)

[Languages](#)

[Topic-wise](#)

[Internships](#)

[Contact Us](#)

[CS Subjects](#)

[How to begin?](#)

[Videos](#)

[Copyright Policy](#)

[Video Tutorials](#)

@geeksforgeeks , Some rights reserved

