**TCP global synchronization**

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/TCP_global_synchronization#mw-head) [Jump to search](https://en.wikipedia.org/wiki/TCP_global_synchronization#searchInput)

|  |  |
| --- | --- |
|  | This article's **tone or style may not reflect the** [**encyclopedic tone**](https://en.wikipedia.org/wiki/Wikipedia:Writing_better_articles#Tone) **used on Wikipedia**. See Wikipedia's [guide to writing better articles](https://en.wikipedia.org/wiki/Wikipedia:Writing_better_articles#Tone) for suggestions. *(December 2008) (*[*Learn how and when to remove this template message*](https://en.wikipedia.org/wiki/Help:Maintenance_template_removal)*)* |

**TCP global synchronization** in [computer networks](https://en.wikipedia.org/wiki/Computer_networks) can happen to [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol)/[IP](https://en.wikipedia.org/wiki/Internet_protocol) flows during periods of congestion because each sender will reduce their transmission rate at the same time when packet loss occurs.

Routers on the Internet normally have packet queues, to allow them to hold packets when the network is busy, rather than discarding them.

Because routers have limited resources, the size of these queues is also limited. The simplest technique to limit queue size is known as [tail drop](https://en.wikipedia.org/wiki/Tail_drop). The queue is allowed to fill to its maximum size, and then any new packets are simply discarded, until there is space in the queue again.

This causes problems when used on [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP) routers handling multiple [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) streams, especially when bursty traffic is present. While the network is stable, the queue is constantly full, and there are no problems except that the full queue results in high latency. However, the introduction of a sudden burst of traffic may cause large numbers of established, steady streams to lose packets simultaneously.

[TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) has automatic recovery from dropped packets, which it interprets as congestion on the network (which is usually correct). The sender reduces its sending rate for a certain amount of time, and then tries to find out if the network is no longer congested by increasing the rate again subject to a ramp-up. This is known as the [slow-start](https://en.wikipedia.org/wiki/Slow-start) algorithm.

Almost all the senders will use the same time delay before increasing their rates. When these delays expire, at the same time, all the senders will send additional packets, the router queue will again overflow, more packets will be dropped, the senders will all back off for a fixed delay... ad infinitum; compare with the [thundering herd problem](https://en.wikipedia.org/wiki/Thundering_herd_problem).

This pattern of each sender decreasing and increasing transmission rates at the same time as other senders is referred to as "global synchronization" and leads to inefficient use of bandwidth, due to the large numbers of dropped packets, which must be retransmitted, and because the senders have a reduced sending rate, compared to the stable state, while they are backed-off, following each loss.

This problem has been the subject of much research. The consensus appears to be that the tail drop algorithm is the leading cause of the problem, and other [queue size management algorithms](https://en.wikipedia.org/wiki/Active_Queue_Management) such as [Random Early Detection (RED)](https://en.wikipedia.org/wiki/Random_early_detection) and [Weighted RED](https://en.wikipedia.org/wiki/Weighted_random_early_detection) will reduce the likelihood of global synchronization, as well as keeping queue sizes down in the face of heavy load and bursty traffic.

**See also**

* [Transmission Control Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) (TCP)
* [TCP congestion avoidance algorithm](https://en.wikipedia.org/wiki/TCP_congestion_avoidance_algorithm)
* [Network congestion](https://en.wikipedia.org/wiki/Network_congestion)
* [Thundering herd problem](https://en.wikipedia.org/wiki/Thundering_herd_problem)

**References**

* *Vegesna, Srinivas (2001). IP Quality of Service, chap. 6. Cisco press.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*1-57870-116-3*](https://en.wikipedia.org/wiki/Special:BookSources/1-57870-116-3)*.*
* *Szigeti, Tim (2005). End-to-End QoS Network Design, chap. 6. Cisco press.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*1-58705-176-1*](https://en.wikipedia.org/wiki/Special:BookSources/1-58705-176-1)*.*

**External links**

* [an ISOC paper discussing the dynamics of TCP, including global synchronization](http://ispcolumn.isoc.org/2005-06/faster.html)
* [A paper with graphs demonstrating TCP global synchronization in action and its impact when TCP coexists with UDP](http://www.isoc.org/inet97/proceedings/F3/F3_1.HTM)