Congestion Window (cwnd) is a TCP state variable that limits the amount of data the [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) can send into the network before receiving an [ACK](https://en.wikipedia.org/wiki/Acknowledgement_(data_networks)). The Receiver Window (rwnd) is a variable that advertises the amount of data that the destination side can receive. Together, the two variables are used to regulate data flow in TCP connections, minimize congestion, and improve network performance.

**Overview**

Network congestion occurs when packets sent from the source (the server) exceed what the destination can handle (the user). When the buffers get filled up at the destination, the packets are temporarily stored in buffers at both the source and destination as they wait to be forwarded to the upper layers. This can lead to congestion and cause packet loss, retransmissions, reduced data throughput, performance degradation, and even network collapse in extreme cases.

TCP can use the congestion window (cwnd) and the receiver window (rwnd) to control and avoid network congestion. The amount of data that can be transmitted through a TCP connection is dependent on the congestion window, which is maintained by the source. The receiver window is maintained by the destination.

**How TCP Uses CWND and RWND**

The destination indicates the size of the receiver window through the [TCP header](http://www.freesoft.org/CIE/Course/Section4/8.htm). This advertises the amount of data that it can receive and enables the congestion window to determine the amount of data it can reliably transmit without an ACK. Once received, the data is stored in the receiver buffer and the receiver sends an ACK or a set of ACKs depending on the amount of received data.

The congestion window size keeps increasing up to the maximum receiver window, or until the network reaches its limit. However, the rate of sending data is bound to the congestion window, and even when the receiver window is large, the congestion window may be smaller – especially if the network does not support transmitting data equivalent to the maximum RWND size.

Congestion is detected either by receipt of a duplicate acknowledgement or timeout signal. Once this happens, the TCP sender decreases the send rate by decreasing the congestion window size by a factor determined by the algorithm used. The maximum amount of unacknowledged data that the source can send is the lower of the two windows.

**Example of CWND and RWND**

In [slow start congestion control](https://en.wikipedia.org/wiki/TCP_congestion_control#Slow_start), TCP increases the window’s size rapidly to reach the maximum transfer rate as fast as possible. This self-imposed window size increases as TCP confirms the network’s ability to transmit the data without errors. However, this can only go up to a maximum advertised window (RWND).

In this scenario, the sender uses two variables:

1. Congestion window with an initial value of one maximum segment size (MSS) and
2. the slow start threshold value (ssthresh) with an initial value equal to the receiver window.

The congestion window increases exponentially during congestion control, and lineally during the [congestion avoidance](http://www.mathcs.emory.edu/~cheung/Courses/558a/Syllabus/6-transport/TCP.html). In the example below, slow start occurs when cwnd **<** ssthresh and congestion avoidance occurs when cwnd **>=** ssthresh

If cwnd <= ssthresh then

Each time an Ack is received:

cwnd = cwnd + 1

else /\* cwnd > ssthresh \*/

Each time an Ack is received:

cwnd = cwnd + 1 / [ cwnd ]

endif

**Conclusion**

The rate of transmitting data in a TCP connection is determined by the rate of incoming ACK, which is dependent on the bottleneck in the round-trip path between the sender and the receiver. The bottleneck may be the Internet or the destination. An adaptive congestion window size allows the TCP to be flexible enough to deal with congestion arising from network or receiver issues.

TCP analyzes the path to the destination and adjusts the congestion window size to reach an optimal send rate with minimal packet loss and congestion.