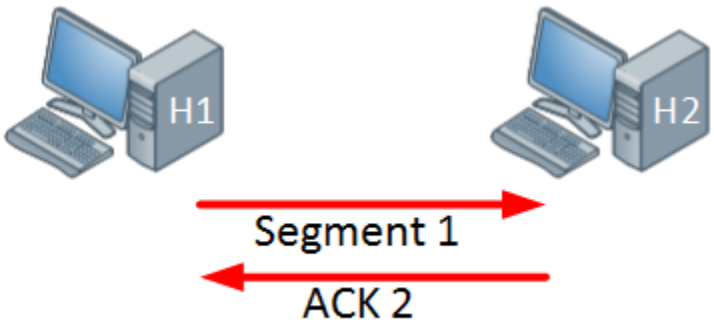
TCP Window Size Scaling

TCP (Transmission Control Protocol) is a connection oriented protocol which means that we keep track of how much data has been transmitted. The sender will transmit some data and the receiver has to acknowledge it. When we don't receive the acknowledgment in time then the sender will re-transmit the data.

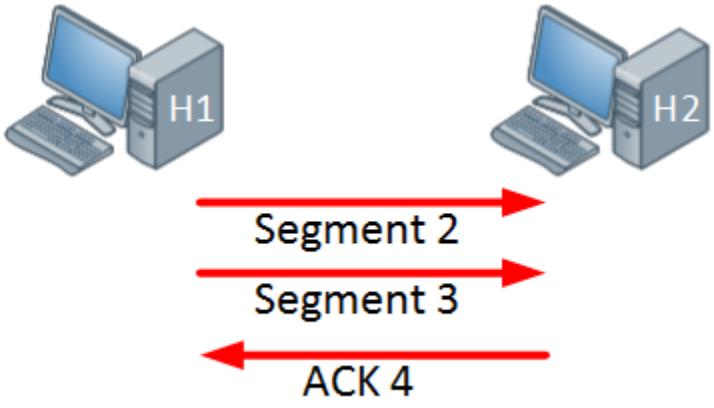
TCP uses "windowing" which means that a sender will send one or more data segments and the receiver will acknowledge one or all segments. When we start a TCP connection, the hosts will use a receive buffer where we temporarily store data before the application can process it.

When the receiver sends an acknowledgment, it will tell the sender how much data it can transmit before the receiver will send an acknowledgment. We call this the **window size**. Basically, the window size indicates the size of the receive buffer.

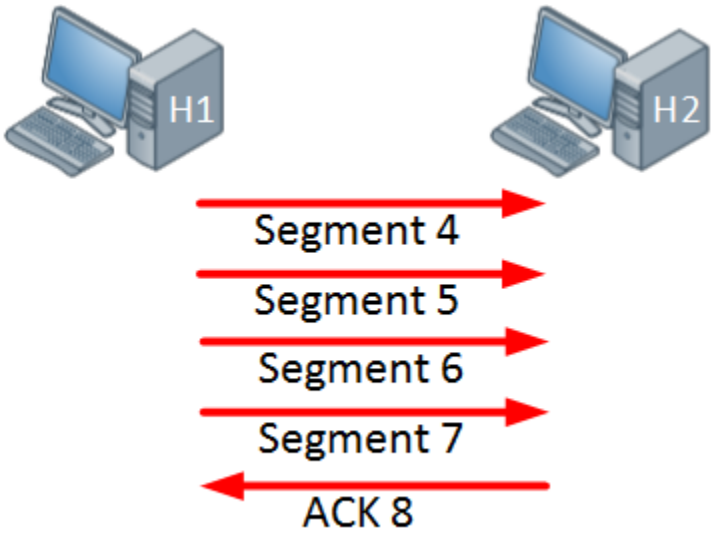
Typically the TCP connection will start with a small window size and every time when there is a successful acknowledgement, the window size will increase. Here's an example:



Above we have two hosts, the host on the left side will send one segment and the host on the right side will send an acknowledgment in return. Since the acknowledgement was successful, the windows size will increase:



The host on the left side is now sending two segments and the host on the right side will return a single acknowledgment. Everything is working fine so the window size will increase even further:



The host is now sending four segments and the host on the right side responds with a single acknowledgment.

In the example above the window size keeps increasing as long as the receiver sends acknowledgments for all our segments or when the window size hits a certain maximum limit. When the receiver doesn't send an acknowledgment within a certain time period (called the round-trip time) then the window size will be reduced.

When an interface has congestion then it's possible that IP packets are dropped. To deal with this, TCP has a number of algorithms that deal with congestion control. One of them is called *slow start*.

With TCP slow start, the window size will initially grow exponentially (window size doubles) but once a packet is dropped, the window size will be reduced to one segment. It will then grow exponentially again until the window size is half of what it was when the congestion occurred. At that moment, the window size will grow linearly instead of exponentially.

When an interface gets congested, it's possible that all your TCP connections will experience TCP slow start. Packets will be dropped and then all TCP connections will have a small window size. This is called **TCP global synchronization**