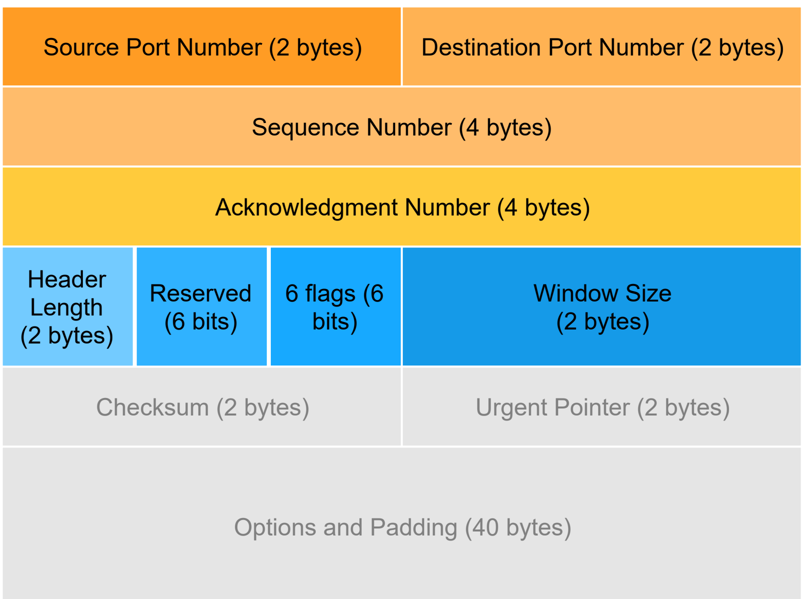
TCP Headers: Window Size, Checksum & More

We'll continue our discussion of TCP headers in this lesson

WE'LL COVER THE FOLLOWING

Window Size

Remember the 'buffer' we discussed in the last lesson? Well, the **window size** is essentially the amount of available space in that buffer. TCP at the receiving end buffers incoming data that has not been processed yet by the overlaying application. The amount of available space in this buffer is specified by the **window size**.

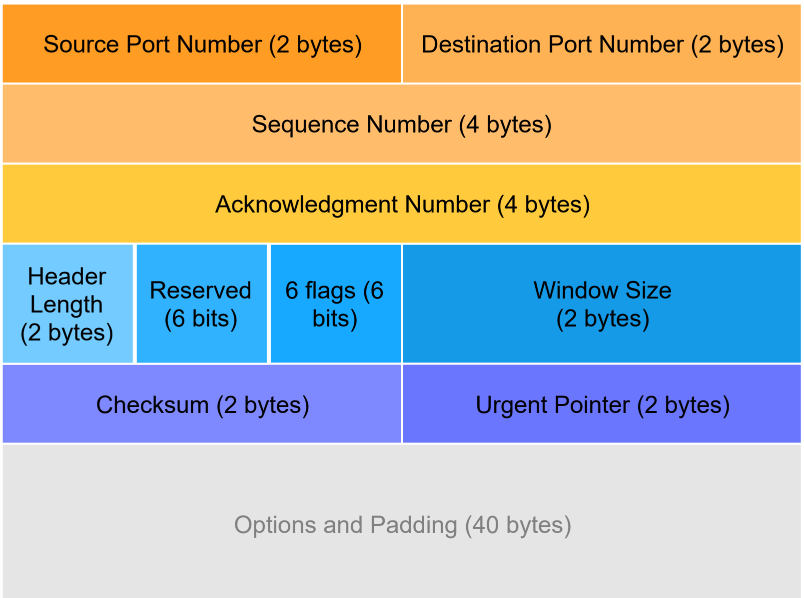


The window size is a critical element of TCP

The window size can be up to **2 bytes** in size hence, the numerical range of the window size is from $2^0 \rightarrow 2^{16} - 1$ or $0 \rightarrow 65535$.

The window size is communicated to the sender by the receiver in every TCP message and gets updated as the buffer fills and empties. If the window size reduces after a bit, the sender will know that it needs to reduce the amount of data being sent, or give the receiver time to clear the buffer.

To put it another way, the window size is at first equal to as much data as the receiving entity is willing and able to receive. As it receives some more data, the window size will decrease and as it hands over some of the received data to the application layer, the window size will increase. This is useful to implement flow control.



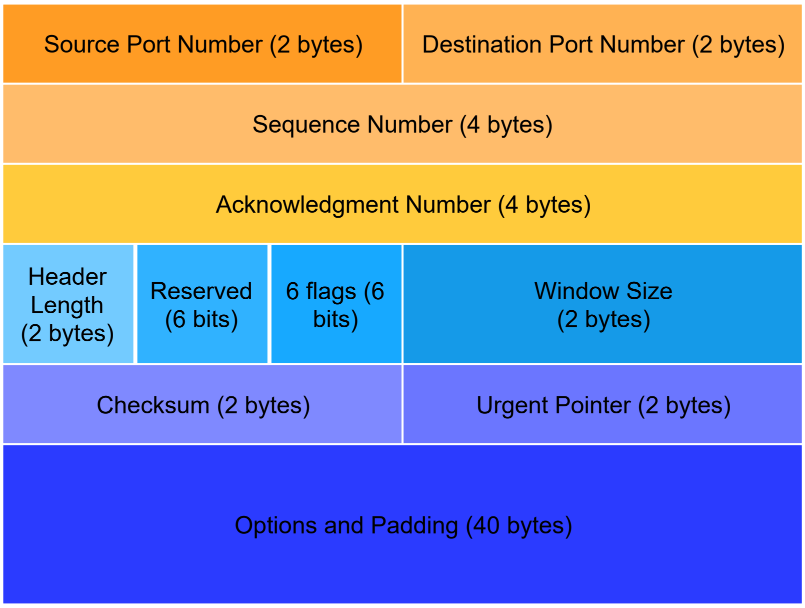
The checksum allows TCP to be reliable, and the urgent pointer identifies the range of bytes in the packet that are urgent.

Checksum

The **checksum** is calculated exactly like in UDP except that the checksum calculation is mandatory in TCP!

Urgent Pointer

The **urgent pointer** defines the byte to the point of which the urgent data exists. This is because a single segment can contain both parts of urgent and regular data. This field is only used in conjunction with the urgent flag.



The options field allows to build in extra functionality that the regular header does not cover

Options & Padding

The **options and padding** field provides up to an extra **40 bytes** to build extra facilities that are not covered by the regular header. The options can vary in length and exist in multiples of 32 bytes using zeros to pad in any extra bits.

Common Options

Some options are commonly used and are well-defined. Here's a table that discusses each.

Option	Explanation
MSS	Defines the maximum-sized payload a host can handle at one time. Larger payload sizes are preferable because the header would be slightly different in certain fields for each segment. The redundancy will be significant and overhead will therefore be high. If this option is not used, the 536-byte payload is used as default.

Timestamp	Allows senders to timestamp segments. This is a logical extension of sequence numbers. We'll have a closer look at this one in upcoming lessons!
Window Scale	Allows the host to 'scale up' its window size by a factor in situations where sending data to the sender from the receiver takes longer than sending to the receiver. This, for example, allows the sender to keep sending data without having to wait long periods of time to hear back for acknowledgments. This option essentially allows the communicating parties to use larger than default buffers - larger by a multiple factor (power of 2 using shift operations). So, if the scale factor is 4 and the window size otherwise specified is 20 kB, the actual window size is 80 kB.

Quick Quiz!

1

The window size represents the amount of buffer available at the receiver. This should be constant and only communicated in the first TCP segment sent from the receiver to the sender. What do you think?

We’re finally done with the TCP header! Let’s learn more about the protocol itself starting from the next lesson.