

1. Source TCP port number (2 bytes)
2. Destination TCP port number (2 bytes)
3. Sequence number (4 bytes)
4. Acknowledgment number (4 bytes)
5. TCP data offset (4 bits)
6. Reserved data (3 bits)
7. Control flags (up to 9 bits)
8. Window size (2 bytes)
9. TCP [checksum](https://www.lifewire.com/what-does-checksum-mean-2625825) (2 bytes)
10. Urgent pointer (2 bytes)
11. TCP optional data (0-40 bytes)

TCP inserts header fields into the message stream in the order listed above.

* *Source and destination TCP port numbers* are the communication endpoints for sending and receiving devices.
* Message senders use *sequence numbers* to mark the ordering of a group of messages. Both senders and receivers use the acknowledgment*numbers* field to communicate the sequence numbers of messages that are either recently received or expected to be sent.
* The *data offset field* stores the total size of a TCP header in multiples of four bytes. A header not using the optional TCP field has a data offset of 5 (representing 20 bytes), while a header using the maximum-sized optional field has a data offset of 15 (representing 60 bytes).
* *Reserved data* in TCP headers always has a value of zero. This field serves the purpose of aligning the total header size as a multiple of four bytes (important for efficiency of computer data processing).
* TCP uses a set of six standard and three extended *control flags*(each an individual bit representing *on* or *off*) to manage data flow in specific situations. One bit flag, for example, initiates TCP connection reset logic. The detailed operation of these fields goes beyond the scope of this article.
* TCP senders use a number called *window size*to regulate how much data they send to a receiver before requiring an acknowledgment in return. If the window size becomes too small, network data transfer will be unnecessarily slow, while if the window size becomes too large, the network link can become saturated (unusable for any other applications) or the receiver may not be able to process incoming data quickly enough (also resulting in slow performance). Windowing algorithms built into the protocol dynamically calculate size values and use this field of TCP headers to coordinate changes between senders and receivers.
* The *checksum*value inside a TCP header is generated by the protocol sender as a mathematical technique to help the receiver detect messages that are corrupted or tampered with.
* The urgent pointer field is often set to zero and ignored, but in conjunction with one of the control flags, it can be used as a data offset to mark a subset of a message as requiring priority processing.
* Usages of optional TCP data go beyond the scope of this article but include support for special acknowledgment and window scaling algorithms.