TCP Header

The TCP header contains 10 mandatory fields, and an optional extension field. The following table is a list of TCP Header fields and their descriptions:

TCP Header Field	Default Action
Source Port	This field contains the sender's port number.
Destination Port	This field is similar to the source port and will contain the port number of the receiver.
Sequence Number	Dual role: 1. If the SYN flag is set, then this is the initial sequence number. The sequence number of the actual first data byte will then be this sequence number plus 1. 2. If the SYN flag is not set, then this is the sequence number of the first data byte.
Acknowledgement Number	If the ACK flag is set, then the value of this field is the next sequence number that the receiver is expecting. A SYN packet should have this value set to 0.
Data Offset (4 bits)	Specifies the size of the TCP header in 32-bit words (multiples of 32-bit chunks). The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.
Reserved (4 bits)	For future use and should be set to zero. **Note:** RFC 3168 (The Addition of Explicit Congestion Notification (ECN) to IP) has taken two bits from the Reserved field and added them to the flags field.

TCP Flags (8 bits)	**URG** (1 bit) – Indicates that the URGent pointer field is significant. **ACK** (1 bit) – Indicates that the ACKnowledgment field is significant. All packets after the initial SYN packet sent by the client should have this flag set. **PSH** (1 bit) – Push function. Asks to push the buffered data to the receiving application. **RST** (1 bit) – Reset the connection. **SYN** (1 bit) – Synchronize sequence numbers. Only the first packet sent from each end should have this flag set. Some other flags change meaning based on this flag, and some are only valid for when it is set, and others when it is clear. **FIN** (1 bit) – No more data from sender. **CWR** (1 bit) – Congestion Window Reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set. **ECE** (ECN-Echo) (1 bit) – Indicates that the TCP peer is ECN capable during 3-way handshake. **Note:** ECN allows end-to-end notification of network congestion without dropping packets. It is an optional feature and is only used when both endpoints signal that they want to use it. When ECN is successfully negotiated, an ECN-aware router may set a bit in the IP header instead of dropping a packet in order to signal the beginning of congestion. The receiver of the packet echoes the congestion indication to the sender, which must react as though a packet drop were detected.
Window (16 bits)	The size of the receive window, which specifies the number of bytes that the receiver is currently willing to receive. For more efficient use of high bandwidth networks, a larger TCP window size may be used (through the window scale option). The TCP window size field controls the flow of data and its value is limited to between 2 and 65,535 bytes. Since the size field cannot be expanded, a scaling factor is used. The TCP window scale option, as defined in RFC 1323, is an option used to increase the maximum window size from 65,535 bytes to 1 gigabyte. Scaling up to larger window sizes is a part of what is necessary for TCP Tuning. The window scale option is used only during the TCP 3-way handshake. The window scale value represents the number of bits to left-shift the 16-bit window size field. The window scale value can be set from 0 (no shift) to 14 for each direction independently. Both sides must send the option in their SYN segments to enable window scaling in either direction. Some routers and packet firewalls rewrite the window scaling factor during a transmission. This causes sending and receiving sides to assume different TCP window sizes. The result is non-stable traffic that may be very slow. The problem is visible on some sites behind a defective router.
Checksum (16 bits)	The 16-bit checksum field is used for error-checking of the header and data.
Urgent pointer (16 bits)	If the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte.

TCP Options (Variable 0-320 bits)	TCP Options (Variable 0-320 bits): The total length of the option field must be a multiple of a 32-bit word and the data offset field adjusted appropriately. Options have up to three fields: Option-Kind (1 byte), Option-Length (1 byte), and Option-Data (variable). The Option-Kind field indicates the type of option and is the only field that is required. Depending on what kind of option we are dealing with, the next two fields may be set: the Option-Length field indicates the total length of the option, and the **Option-Data** field contains the value of the option, if applicable. For example, an Option-Kind byte of 0x01 indicates that this is a No-Op option used only for padding, and does not have an Option-Length or Option-Data byte following it. An Option-Kind byte of 0 is the End Of Options option and is also only one byte. An Option-Kind byte of 0x02 indicates that this is the Maximum Segment Size option and will be followed by a byte specifying the length of the MSS field (should be 0x04). Note that this length is the total length of the given options field, including Option-Kind and Option-Length bytes. So while the MSS value is typically expressed in two bytes, the length of the field will be 4 bytes (+2 bytes of kind and length). In short, an MSS option field with a value of 0x05B4 will show up as (0x02 0x04 0x05B4) in the TCP options section. Some options may only be sent when SYN is set; they are indicated below as **[SYN]**. Option-Kind and standard lengths given as (Option-Kind, Option-Length). • **0** (8 bits): End of options list. • **1** (8 bits): No operation (NOP, Padding). This may be used to align option fields on 32-bit boundaries for better performance. • **2,4,SS** (32 bits):– Maximum segment size **[SYN]** • **3,3,S** (24 bits): Window scale **[SYN]** • **4,2** (16 bits): Selective Acknowledgement permitted (allows the receiver to acknowledge discontinuous blocks of packets which were received correctly). **[SYN]** • **5,N,BBBB,EEEE,...** (variable bits, N is either 10, 18, 26, or 34): Selective ACKnowledgement (SACK). These first two bytes are followed by a list of 1–4 blocks being selectively acknowledged, specified as 32-bit begin/end pointers. • **8,10,TTTT,EEEE** (80 bits): Timestamp and echo of previous timestamp. TCP timestamps, defined in RFC 1323, can help TCP determine in which order packets were sent. TCP timestamps are not normally aligned to the system clock and start at some random value. Many operating systems will increment the timestamp for every elapsed millisecond; however the RFC only states that the ticks should be proportional. • **14,3,S** (24 bits): TCP Alternate Checksum Request. **[SYN]** • **15,N,...** (variable bits): TCP Alternate Checksum Data.

Recommended Internet Sites

- RFC 793: https://web.archive.org/web/20160718145722/https://tools.ietf.org/html/rfc793
- RFC 1340: https://web.archive.org/web/20160718145802/https://tools.ietf.org/html/rfc1340
- TCP and UDP port numbers: https://web.archive.org/web/20160718145857/https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Please contact the Course Coordinators if you are unable to access any of the Recommended Internet Sites.