3-way Handshake

Key Features of TCP-

Connection oriented

Point-to-point communication: Two end-points

Reliable transfer: Data is delivered in order

Full duplex communication

Stream interface: Continuous sequence of octets

Reliable connection startup: Data on old connection does not confuse new connections

Graceful connection shutdown: Data sent before closing a connection is not lost.

The Transmission Control Protocol is the most common transport layer protocol. It works together with IP and provides a reliable transport service between processes using the network layer service provided by the IP protocol.
The various **services** provided by the TCP to the application layer are as follows:

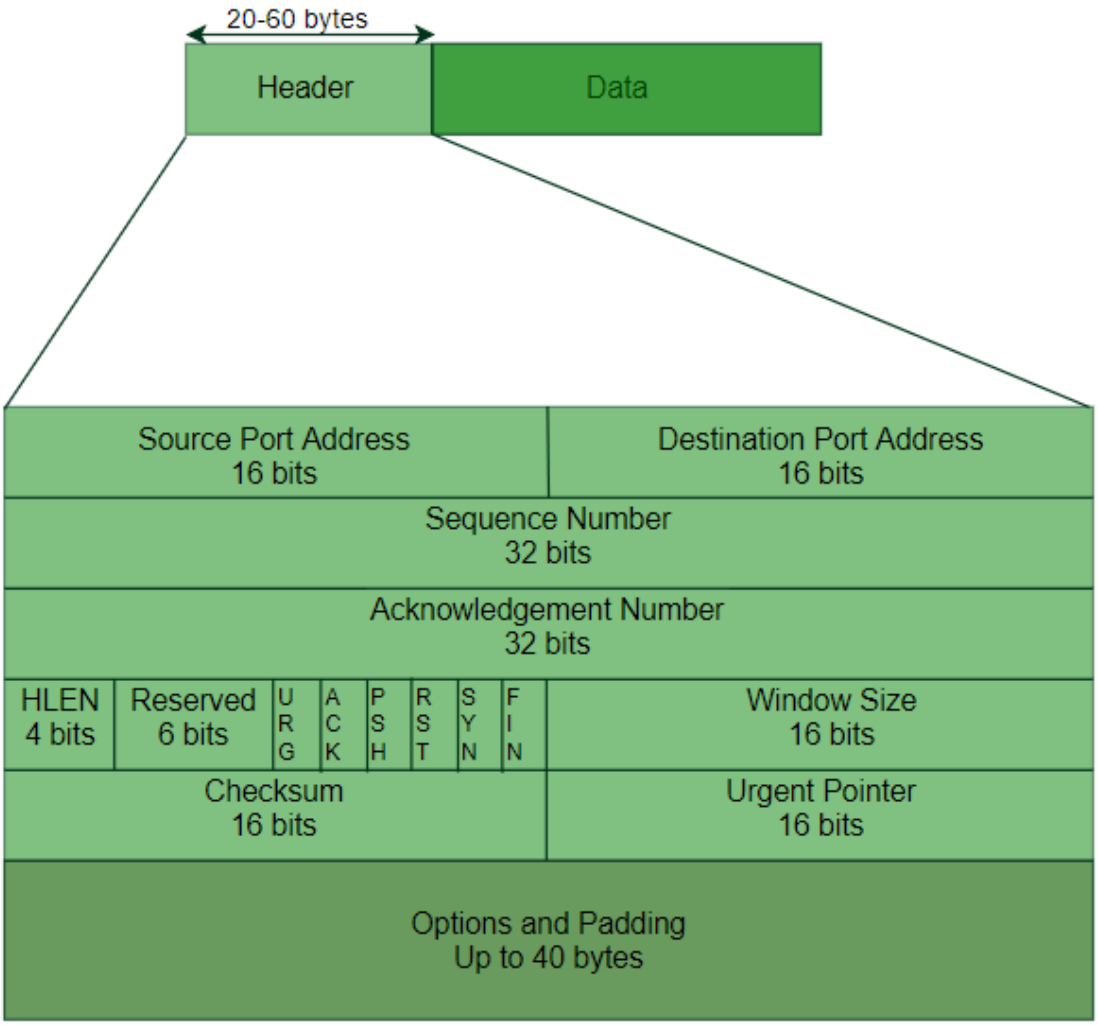
1. **Process-to-Process Communication –**
   TCP provides process to process communication, i.e, the transfer of data takes place between individual processes executing on end systems. This is done using port numbers or port addresses. Port numbers are 16 bit long that help identify which process is sending or receiving data on a host.
2. **Stream oriented –**
   This means that the data is sent and received as a stream of bytes(unlike UDP or IP that divides the bits into datagrams or packets). However, the network layer, that provides service for the TCP, sends packets of information not streams of bytes. Hence, TCP groups a number of bytes together into a *segment* and adds a header to each of these segments and then delivers these segments to the network layer. At the network layer, each of these segments are encapsulated in an IP packet for transmission. The TCP header has information that is required for control purpose which will be discussed along with the segment structure.
3. **Full duplex service –**
   This means that the communication can take place in both directions at the same time.
4. **Connection oriented service –**
   Unlike UDP, TCP provides connection oriented service. It defines 3 different phases:
   - Connection establishment
   - Data transfer
   - Connection termination

   (IMP: This is a virtual connection, not a physical connection, means during the transmission the resources will not be reserved and the segments will not follow the same path to reach the destination but it is a connection orientation in the sense that segments will arrive in order by the help of sequence number.)
5. **Reliability –**

TCP is reliable as it uses checksum for error detection, attempts to recover lost or corrupted packets by re-transmission, acknowledgement policy and timers. It uses features like byte number and sequence number and acknowledgement number so as to ensure reliability. Also, it uses congestion control mechanisms.

6. **Multiplexing –**
   TCP does multiplexing and de-multiplexing at the sender and receiver ends respectively as a number of logical connections can be established between port numbers over a physical connection.

TCP Segment structure –
TCP segment consists of data bytes to be sent and a header that is added to the data by TCP as shown:



The header of a TCP segment can range from 20-60 bytes. 40 bytes are for options. If there are no options, header is of 20 bytes else it can be of upmost 60 bytes.
Header fields:

- **Source Port Address –**
  16 bit field that holds the port address of the application that is sending the data segment.
- **Destination Port Address –**

16 bit field that holds the port address of the application in the host that is receiving the data segment.

- **Sequence Number –**
  32 bit field that holds the sequence number, i.e, the byte number of the first byte that is sent in that particular segment. It is used to reassemble the message at the receiving end if the segments are received out of order.
- **Acknowledgement Number –**
  32 bit field that holds the acknowledgement number, i.e, the byte number that the receiver expects to receive next. It is an acknowledgment for the previous bytes being received successfully.
- **Header Length (HLEN) –**
  This is a 4 bit field that indicates the length of the TCP header by number of 4-byte words in the header, i.e, if the header is of 20 bytes(min length of TCP header), then this field will hold 5 (because 5 x 4 = 20) and the maximum length: 60 bytes, then it'll hold the value 15(because 15 x 4 = 60). Hence, the value of this field is always between 5 and 15.
- **Control flags –**
  These are 6 1-bit control bits that control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc. Their function is:

    - URG: Urgent pointer is valid. Data inside a segment with URG = 1 flag is forwarded to application layer immediately even if there are more data to be given to application layer. It is used to notify the receiver to process the urgent packets before processing all other packets. The receiver will be notified when all known urgent data has been received.
    - ACK: Acknowledgement number is valid (used in case of cumulative acknowledgement). It is used to acknowledge packets which are successful received by the host. The flag is set if the acknowledgement number field contains a valid acknowledgement number.
      In given below diagram, the receiver sends an ACK = 1 as well as SYN = 1 in the second step of connection establishment to tell sender that it received its initial packet.
    - PSH: Request for push. Transport layer by default waits for some time for application layer to send enough data equal to maximum segment size so that the number of packets transmitted on network minimizes which is not desirable by some application like interactive applications(chatting). Similarly transport layer at receiver end buffers packets and transmit to application layer if it meets certain criteria.

This problem is solved by using PSH. Transport layer sets PSH = 1 and immediately sends the segment to network layer as soon as it receives signal from application layer. Receiver transport layer, on seeing PSH = 1 immediately forwards the data to application layer.
In general, it tells the receiver to process these packets as they are received instead of buffering them.

- RST: Reset the connection. It is used to terminate the connection if the RST sender feels something is wrong with the TCP connection or that the conversation should not exist. It can get send from receiver side when packet is send to particular host that was not expecting it.
- SYN: Synchronize sequence numbers. It is used in first step of connection establishment phase or 3-way handshake process between the two hosts.

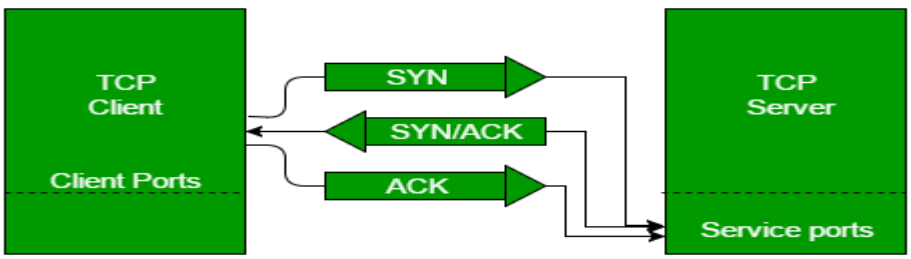
Only the first packet from sender as well as receiver should have this flag set. This is used for synchronizing sequence number i.e. to tell the other end which sequence number they should except.

- FIN: Terminate the connection. It is used to request for connection termination i.e. when there is no more data from the sender, it requests for connection termination. This is the last packet sent by sender. It frees the reserved resources and gracefully terminate the connection.

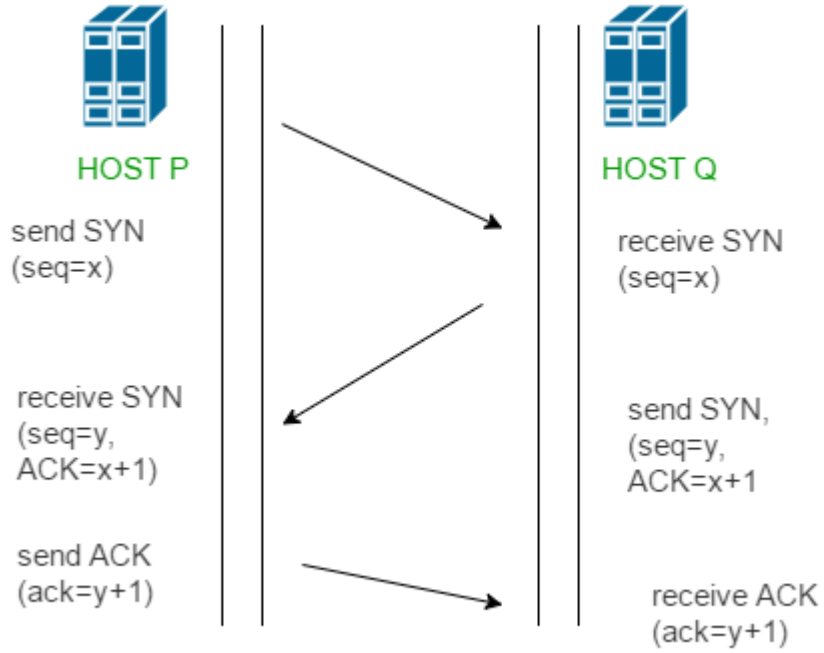
- **Window size –**
  This field tells the window size of the sending TCP in bytes.
- **Checksum –**
  This field holds the checksum for error control. It is mandatory in TCP as opposed to UDP.
- **Urgent pointer –**
  This field (valid only if the URG control flag is set) is used to point to data that is urgently required that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte.

TCP Connection –
TCP is connection oriented. A TCP connection is established by 3-way handshake.

The process of communication between devices over the internet happens according to the current **TCP/IP** suite model(stripped out version of OSI reference model). The Application layer is a top pile of stack of TCP/IP model from where network referenced application like web browser on the client side establish connection with the server. From the application layer,the information is transferred to the transport layer where our topic comes into picture. The two important protocols of this layer are – TCP, **UDP(User Datagram Protocol)** out of which TCP is prevalent(since it provides reliability for the connection established). However you can find application of UDP in querying the DNS server to get the binary equivalent of the Domain Name used for the website.



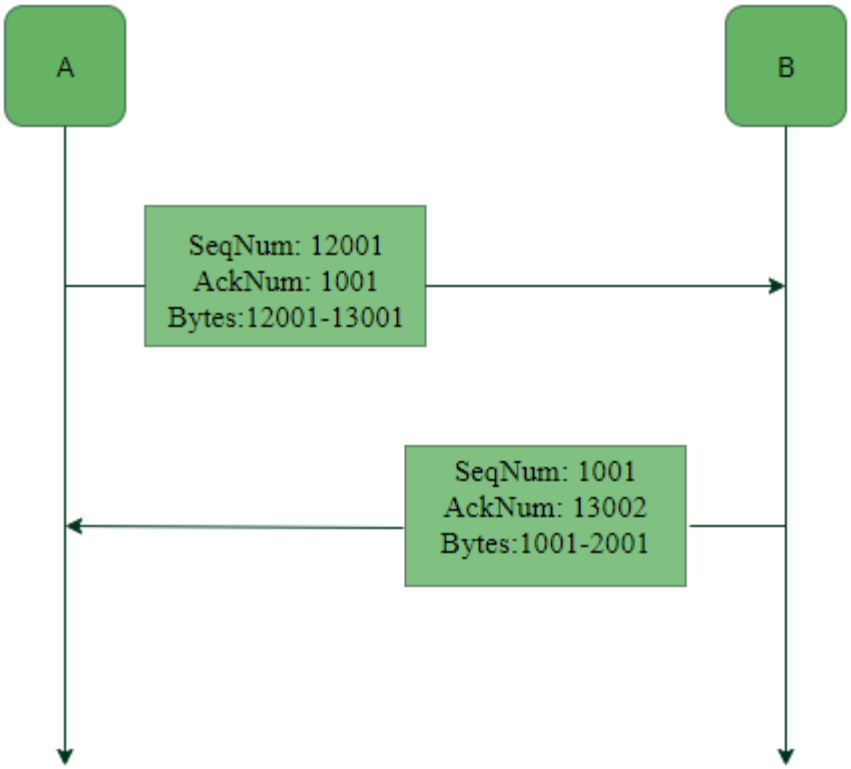
TCP provides reliable communication with something called **Positive Acknowledgement with Re-transmission(PAR)**. The Protocol Data Unit(PDU) of the transport layer is called segment. Now a device using PAR resend the data unit until it receives an acknowledgement. If the data unit received at the receiver's end is damaged(It checks the data with checksum functionality of the transport layer that is used for Error Detection), then receiver discards the segment. So the sender has to resend the data unit for which positive acknowledgement is not received. You can realize from above mechanism that three segments are exchanged between sender(client) and receiver(server) for a reliable TCP connection to get established. Let us delve how this mechanism works :



- **Step 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer

The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

Byte number, Sequence number and Acknowledgement number:
All the data bytes that are to be transmitted are numbered and the beginning of this numbering is arbitrary. Sequence numbers are given to the segments so as to reassemble the bytes at the receiver end even if they arrive in a different order. Sequence number of a segment is the byte number of the first byte that is being sent. Acknowledgement number is required since TCP provides full duplex service. Acknowledgement number is the next byte number that the receiver expects to receive which also provides acknowledgement for receiving the previous bytes. Example:



In this example we see that, A sends acknowledgement number1001, which means that it has received data bytes till byte number 1000 and expects to receive 1001 next, hence B next sends data bytes starting from 1001. Similarly, since B has received data bytes till byte number 13001 after the first data transfer from A to B, therefore B sends acknowledgement number 13002, the byte number that it expects to receive from A next.