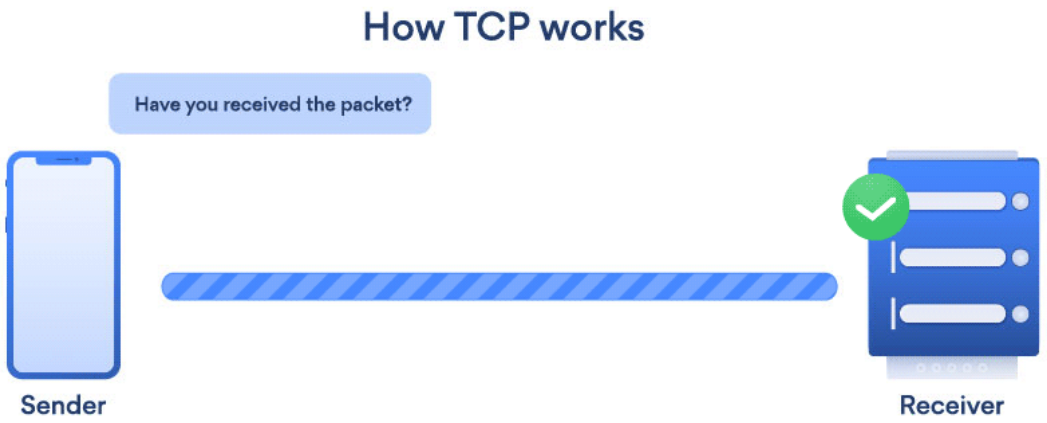
Transmission Control Protocol (TCP) is a connection-oriented protocol for sending data or packets of information over the internet. This means once a connection is established, data can flow in both directions.

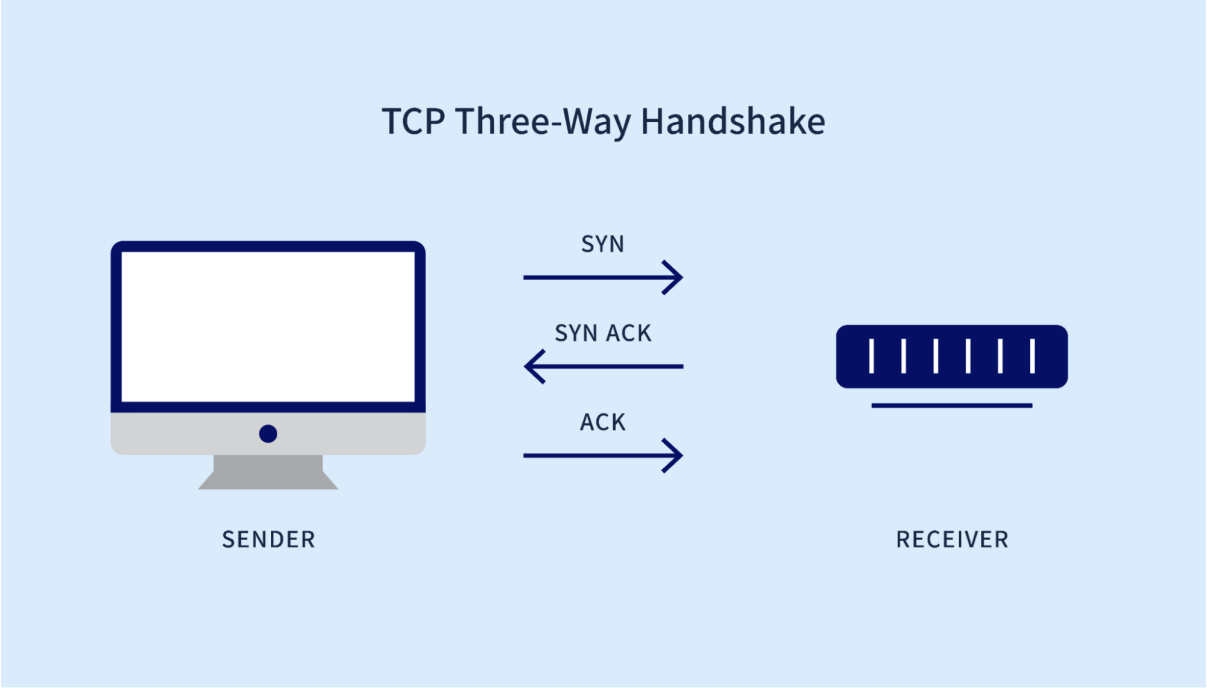


This method of delivery is extremely reliable, as it guarantees data delivery to the destination server. There are also built-in systems that check for errors and ensure that data gets sent in the correct order.

TCP is well-suited for transferring most data types (e.g., webpages, emails, documents, etc.) over the internet, where reliability is imperative.

How does TCP work?

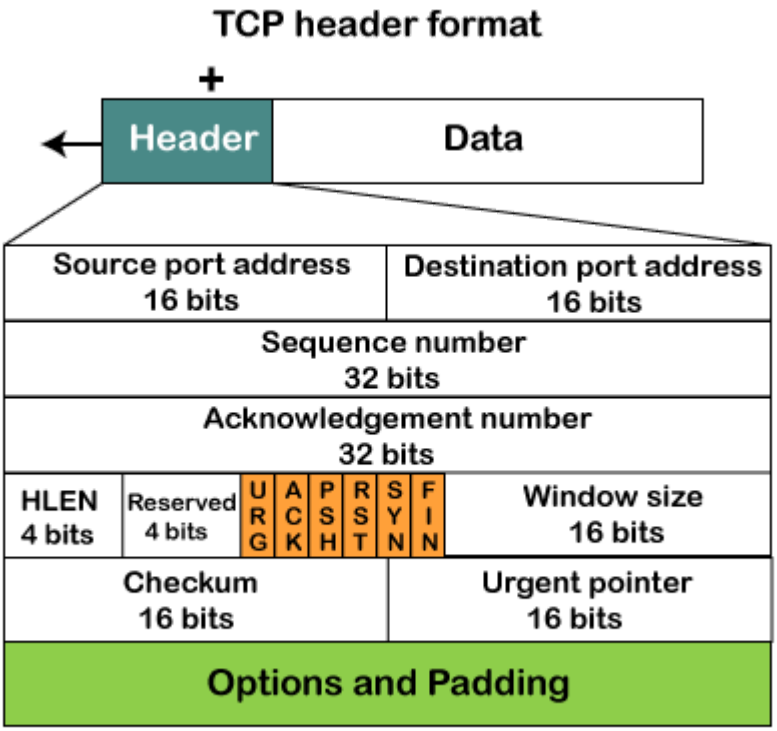
TCP is the most commonly used protocol for sending and receiving data over the internet. It’s not exactly the fastest, but it’s more reliable than its UDP counterpart (more on that later).



Here's how the TCP protocol works:

1. Connections start with what's called a "three-way handshake" — a three-step process that involves exchanging synchronization and acknowledgment packets before data gets transmitted.
2. Once the TCP handshake is complete, you can get to work on your email (or another task). Your data gets divided into smaller units called "packets" or diagrams. These are part of a larger message.
3. A unique identifier and a sequence number are assigned to each TCP packet. These numbers guarantee that the data packets are sent and received in the correct order.
4. When you send data over TCP, the receiver sends an acknowledgment back to the sender (if it's in the correct order). The sequence and acknowledgment numbers are used to keep track of individual data packets.
5. Data packets can get lost or arrive in the wrong order when traveling across a computer network. If the sender doesn't receive an acknowledgment, the data packet gets sent again. If the data is sent in the wrong order, the recipient can use the sequence numbers to reassemble the data.
6. Finally, either side can send a FIN packet to close the TCP connection.

TCP Header format



- **Source port:** It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.
- **Destination port:** It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.
- **Sequence number:** This field contains the sequence number of data bytes in a particular session.
- **Acknowledgment number:** When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.
- **HLEN:** It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.
- **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.

- **Flags**

  There are six control bits or flags:

  1. **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
  2. **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.
  3. **PSH:** If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
  4. **RST:** If it is set, then it requests to restart a connection.
  5. **SYN:** It is used to establish a connection between the hosts.
  6. **FIN:** It is used to release a connection, and no further data exchange will happen.

- **Window size**

  It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.

- **Checksum**

  It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.

- **Urgent pointer**

  It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.

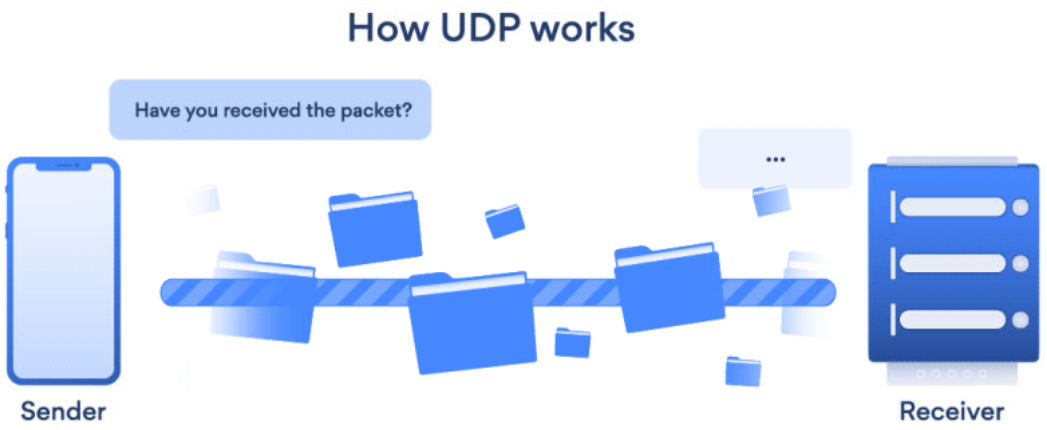
- **Options**

  It provides additional options. The optional field is represented in 32-bits. If

this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

What is UDP?

User Datagram Protocol (UDP) is a connectionless protocol. It doesn't require a "handshake," and data packets are sent in a continuous stream. This makes data transfers much faster than TCP.



UDP is also "lighter" than TCP. It has practically no overhead, and there's no additional processing aside from sending the actual data.

This makes UDP ideal for online activities like video streaming, online gaming, and live broadcasts. Speed is more important than accuracy for these types of communications.

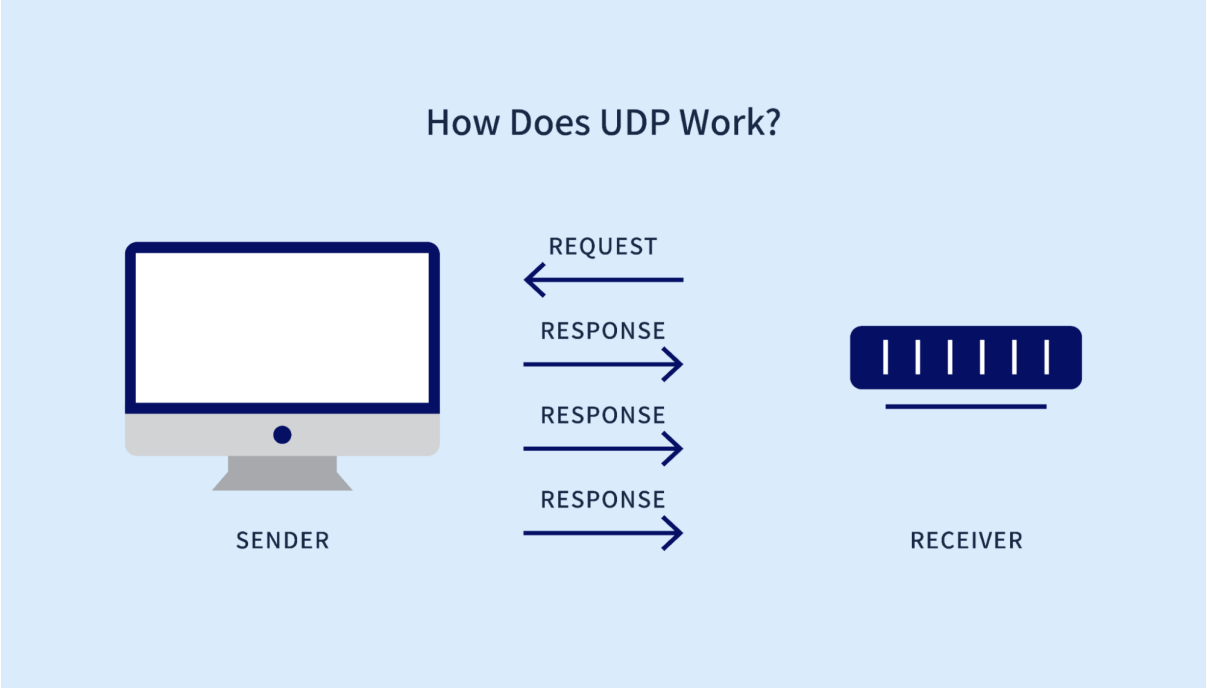
While UDP is faster than TCP, it's not as reliable. Some data packets may be lost during transfers, and there are no mechanisms in place for retransmissions. If you're sending something important, then you'll want to think twice about using this protocol.

How does UDP work?

Data transfers are more straightforward with UDP.

TCP connections always start with a three-way handshake to synchronize and acknowledge data packets. This guarantees data transfers.

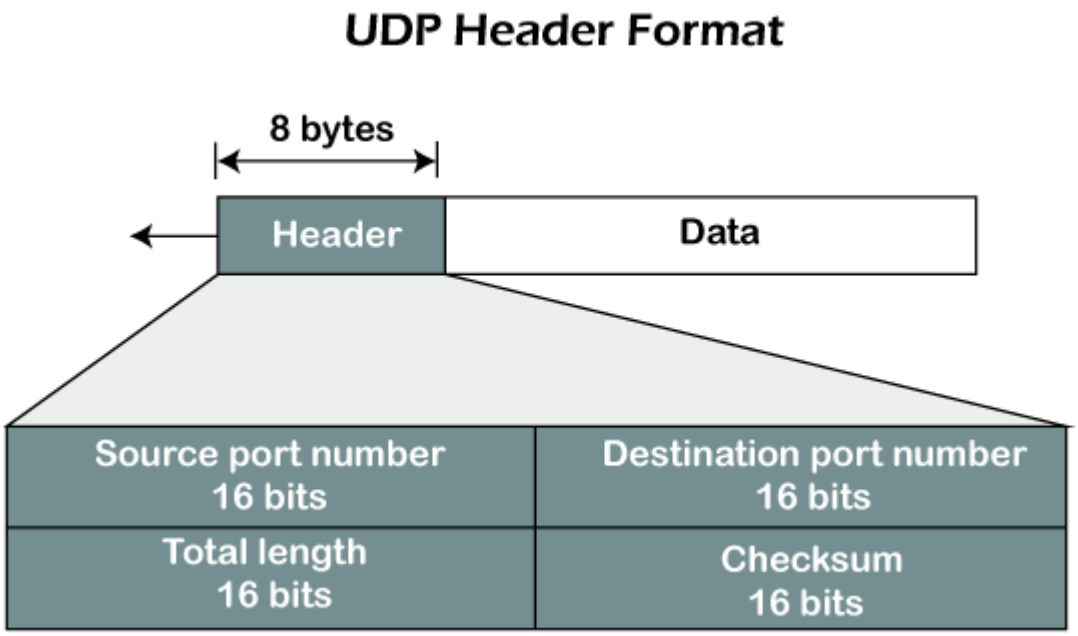
UDP communications don't need to go through this process. Data packets are sent directly to a target device without having to establish a connection or check the order of said packets.



UDP has what's called a "checksum" — a mechanism that checks for corrupted data packets. Here's how it works:

1. The sender sends a checksum value (the number of bits in a message).
2. The receiver also calculates the checksum. If the values match, the data is uncorrupted.
3. If the checksum values *don't* match, the receiver knows the data is corrupted. Lost or corrupted data packets are simply discarded.

UDP Header Format



In UDP, the header size is 8 bytes, and the packet size is upto 65,535 bytes. But this packet size is not possible as the data needs to be encapsulated in the IP datagram, and an IP packet, the header size can be 20 bytes; therefore, the maximum of UDP would be 65,535 minus 20. The size of the data that the UDP packet can carry would be 65,535 minus 28 as 8 bytes for the header of the UDP packet and 20 bytes for IP header.

The UDP header contains four fields:

- **Source port number:** It is 16-bit information that identifies which port is going t send the packet.
- **Destination port number:** It identifies which port is going to accept the information. It is 16-bit information which is used to identify application-level service on the destination machine.
- **Length:** It is 16-bit field that specifies the entire length of the UDP packet that includes the header also. The minimum value would be 8-byte as the size of the header is 8 bytes.
- **Checksum:** It is a 16-bits field, and it is an optional field. This checksum field checks whether the information is accurate or not as there is the possibility that the information can be corrupted while transmission. It is an optional field,

which means that it depends upon the application, whether it wants to write the checksum or not. If it does not want to write the checksum, then all the 16 bits are zero; otherwise, it writes the checksum. In UDP, the checksum field is applied to the entire packet, i.e., header as well as data part whereas, in IP, the checksum field is applied to only the header field.

TCP is ideal for:

- Browsing the web
- Sending emails
- Exchanging files

UDP is ideal for:

- Live streaming
- Video conferencing
- Online gaming

	TCP	UDP
Full form	It stands for **Transmission Control Protocol**.	It stands for **User Datagram Protocol**.
Type of connection	It is a connection-oriented protocol, which means that the connection needs to be established before the data is transmitted over the network.	It is a connectionless protocol, which means that it sends the data without checking whether the system is ready to receive or not.

Reliable	TCP is a reliable protocol as it provides assurance for the delivery of data packets.	UDP is an unreliable protocol as it does not take the guarantee for the delivery of packets.
Speed	TCP is slower than UDP as it performs error checking, flow control, and provides assurance for the delivery of	UDP is faster than TCP as it does not guarantee the delivery of data packets.
Header size	The size of TCP is 20 bytes.	The size of the UDP is 8 bytes.
Acknowledgment	TCP uses the three-way-handshake concept. In this concept, if the sender receives the ACK, then the sender will send the data. TCP also has the ability to resend the lost data.	UDP does not wait for any acknowledgment; it just sends the data.
Flow control mechanism	It follows the flow control mechanism in which too many packets cannot be sent to the receiver at the same time.	This protocol follows no such mechanism.
Error checking	TCP performs error checking by using a checksum. When the	It does not perform any error checking, and also does not resend the lost data packets.

	data is corrected, then the data is retransmitted to the receiver.	
Applications	This protocol is mainly used where a secure and reliable communication process is required, like military services, web browsing, and e-mail.	This protocol is used where fast communication is required and does not care about the reliability like VoIP, game streaming, video and music streaming, etc.