The **Transmission Control Protocol (TCP)** is a transport protocol that is used on top of IP to ensure reliable transmission of packets.

TCP includes mechanisms to solve many of the problems that arise from packet-based messaging, such as lost packets, out of order packets, duplicate packets, and corrupted packets.

Since TCP is the protocol used most commonly on top of IP, the Internet protocol stack is sometimes referred to as **TCP/IP**.

**Packet format**

When sending packets using TCP/IP, the data portion of each IP packet is formatted as a **TCP segment**.

Graphical user interface

Description automatically generated

Each TCP segment contains a header and data. The TCP header contains many more fields than the UDP header and can range in size from 20 to 60 bytes, depending on the size of the options field.

The TCP header shares some fields with the UDP header: source port number, destination port number, and checksum.

### **Step 1: Establish connection**

When two computers want to send data to each other over TCP, they first need to establish a connection using a **three-way handshake**.

The first computer sends a packet with the SYN bit set to 1 (SYN = "synchronize?"). The second computer sends back a packet with the ACK bit set to 1 (ACK = "acknowledge!") plus the SYN bit set to 1. The first computer replies back with an ACK.

In fact, the **three packets** involved in the three-way handshake do not typically include any data. Once the computers are done with the handshake, they're ready to receive packets containing actual data.

### **Step 2: Send packets of data**

When a packet of data is sent over TCP, the recipient must always acknowledge what they received.

The first computer sends a packet with data and a sequence number. The second computer acknowledges it by setting the ACK bit and increasing the acknowledgement number by the length of the received data.

### **Step 3: Close the connection**

Either computer can close the connection when they no longer want to send or receive data.

A computer initiates closing the connection by sending a packet with the FIN bit set to 1 (FIN = finish). The other computer replies with an ACK and another FIN. After one more ACK from the initiating computer, the connection is closed.

**Detecting lost packets**

TCP connections can detect lost packets using a timeout.

After sending off a packet, the sender starts a timer and puts the packet in a retransmission queue. If the timer runs out and the sender has not yet received an ACK from the recipient, it sends the packet again.

The retransmission may lead to the recipient receiving duplicate packets, if a packet was not actually lost but just very slow to arrive or be acknowledged. If so, the recipient can simply discard duplicate packets. It's better to have the data twice than not at all!

**Handling out of order packets**

TCP connections can detect out of order packets by using the sequence and acknowledgement numbers.

When the recipient sees a higher sequence number than what they have acknowledged so far, they know that they are missing at least one packet in between.

Sometimes the missing packet is simply taking a slower route through the Internet and it arrives soon after.

Other times, the missing packet may actually be a lost packet and the sender must retransmit the packet.

In both situations, the recipient has to deal with out of order packets. Fortunately, the recipient can use the sequence numbers to reassemble the packet data in the correct order.