

<https://stackoverflow.com/questions/23157817/http-vs-tcp-ip-send-data-to-a-web-server>

One might regard TCP or UDP as the equivalent of the US Postal Service, while HTTP is the equivalent of one business letter template. If you are writing to an unknown person, as business letter format is a good, generic way to transmit the "WHO, HOW, WHAT, WHEN, AND WHERE" of your message, but it is NOT the only format that is allowed by the USPS. That is, if you are writing to a business associate or loved one, you might forgo the formalities and instead go with a more efficient format, like a billing invoice or love letter or greeting card, but the USPS, like TCP or UDP, will still be there for you, carrying the letters back and forth.

In Short: TCP is a transport-layer protocol, and HTTP is an application-layer protocol that runs over TCP.

Detail: To understand the difference (and a lot of other networking topics), you need to understand the idea of a layered networking model. Essentially, there are different protocols that let a computer talk at different distances and different layers of abstraction.

At the very bottom of the network stack is the physical layer. This is where electrical signals or light pulses or radio waves actually transmit information from place to place. The physical layer doesn't really have protocols, but instead has standards for voltages, frequencies, and other physical properties. You can transmit information directly this way, but you need a lot of power or a dedicated line, and without higher layers you won't be able to share bandwidth.

The next layer up is the link layer. This layer covers communication with devices that share a physical communications medium. Here, protocols like Ethernet, 802.11a/b/g/n, and Token Ring specify how to handle multiple concurrent accesses to the physical medium and how to direct traffic to one device instead of another. In a typical home network, this is how your computer talks to your home "router."

The third layer is the network layer. In the majority of cases, this is dominated by Internet Protocol (IP). This is where the magic of the Internet happens, and you get to talk to a computer halfway around the world, without needing to know where it is. Routers handle directing your traffic from your local network to the network where the other computer lives, where its own link layer handles getting the packets to the right computer.

Now we are getting somewhere. We can talk to a computer somewhere around the world, but that computer is running lots of different programs. **It could be a web server which understands HTTP or could be another program that understand SMTP** How should it know which one to deliver your message to? The transport layer takes care of this, usually with port numbers. The two most popular transport layer protocols are TCP and UDP. TCP does a lot of interesting things to smooth over the rough spots of network-layer packet-switched communication like reordering packets, retransmitting lost packets, etc. UDP is more unreliable, but has less overhead.

So we've connected your browser to the web server software on the other end, but how does the server know what page you want? How can you post a question or an answer? These are things that application-layer protocols handle. For web traffic, this is the HyperText Transfer Protocol (HTTP). There are thousands of application-layer protocols: SMTP, IMAP, and POP3 for email; XMPP, IRC, ICQ for chat; Telnet, SSH, RDP for remote administration; etc.

These are the five layers of the TCP/IP networking model, but they are really only conceptual. The OSI model has 7 layers. In reality, some protocols shim between various layers, or can work at multiple layers at once. TLS/SSL for instance provides encryption and session information between the network and transport layers. Above the application layer, Application Programming Interfaces (APIs) govern communication with web applications like Quora, Twitter, and Facebook.