TCP vs UDP

(Layer 4 concept - transport layer - of the OSI model)

Vocabulary:

1. Transmission Control Protocol (TCP)
   1. *Connection-full* protocol, which means it’s a reliable way to transport segments across the network
      1. 3-way Handshake, SYN-SYNACK-ACK
         1. Tells us if a successful 2 way communication was established with careful checking and acknowledging; great communication. “Certified mail”.
2. User Datagram Protocol (UDP)
   1. Unreliable and it transmits segments called datagrams. UDP is a *connectionless protocol* (doesn’t have to wait for connections).
      1. Not great communication like TCP; if UDP is sending datagrams, and some get dropped, the sender will never even know that it happened.
      2. UDP is really good for audio and visual streaming; because you send a lot of data, and you don’t have the ‘checks and balances’ system like TCP, which could cause certain applications to buffer, like streaming video, because if one frame gets dropped, you won’t notice. So UDP it is!

| **TCP** | **UDP** |
| --- | --- |
| Reliable (three-way handshake) | Not reliable |
| Connection-oriented | Connectionless |
| Segment retransmission and flow control (windowing) | No retransmission and no windowing |
| With segmentation of sequencing | Without sequencing |
| With acknowledgement | Without acknowledgement |

| **TCP** | **UDP** |
| --- | --- |
| Reliable (three-way handshake) | Not reliable, no three-way handshake |
| Connection-oriented, Connection-full protocol | Connectionless, “fire and forget method” |
| Segment retransmission and flow control (windowing) | No retransmission and no windowing |
| With segmentation of sequencing, when you send out data in order, like 1-100, and they get jumbled during transmission, TCP will recorder for you | Without sequencing, UDP will not do the thing TCP does, lol. |
| With acknowledgement, “If I didn’t get it, I know I didn’t get it, and I can get it retransmitted to me.” | Without acknowledgement, a lot less overhead |

* TCP (Connection-Oriented)
  + SSH, HTTP, or HTTPs
* UDP (Connectionless)
  + DHCP, TFTP, audio/video streaming
    - For example, DHCP sends out a broadcast message, but if no one responds, it will simply try again. Then when no one responds, it will try again. Then if no one responds, it will try - do you see the problem?