Congestion control in our Project :

In our proxy server code, it handles this through the congestion window (congestion_window) in the client_info dictionary for each client. When a new client connects, its congestion window is initialized to 1 ('congestion_window': 1). Each time a valid packet is received from a client, the congestion window is increased by 1 (client_info[addr]['congestion_window'] += 1). When an acknowledgment (ACK) is sent to the client, the congestion window is reduced by 1 (client_info[addr]['congestion_window'] -= 1). If the congestion window reaches the maximum size (max_window), a SIGNFULL signal is sent to the client (send_signfull_signal(server, addr)). This mechanism makes it possible to control congestion by adapting the congestion window according to the quantity of packets in transit. When the congestion window is small, it means that the server sends fewer packets, which reduces the load on the network and limits congestion. When the congestion window is large, it means that the server sends more packets, which can increase the load on the network and potentially cause congestion.

For the client: When it receives a "SIGNFULL" signal from the server, the client pauses sending packets for 10 seconds (time.sleep(10)) before resuming. This gives the server time to free up space in its receive window.