Sorry for making a document, I hit the character limit for Discord pretty quickly

Okay, so here is the issue in more detail. Typically whenever someone fills out any kind of form online the data is all stored in the browser until the user clicks submit. Once that happens a single request is sent to the server containing all the data. So if the data is 5KB, then that's how much is sent (slightly more with headers but that's pretty nominal for a single request) I decided to look closer at the socket data.

When I was inspecting the socket traffic while making a post, I found that a request was made to the server *for every character I typed*. Whenever I looked at the data, I noticed that it wasn't just the new character I had typed, it was the entire post that was sent repeatedly with each character I typed. Because the entire message is sent each time the performance gets worse the longer the post is. Also, backspaces send requests too, so there is no limit to how many requests could be sent for an individual post.

As you can see at the bottom of the screenshot the console tracks how much total data has been sent for that socket session. I copied and pasted a 4,000 character post. Since it was copy paste, all the data was sent in one request, which measured to be about 128KB. Then I typed out a 4,000 character message one character at a time (as a user would) to send and measured how much data was going over

the wire. What I found was that when individual characters are typed, it sent over 18.5MB (19,398,656KB) over the wire!!

This may not sound like much for one user but that is 151,552 (19,398,656KB/128KB) times more data than would have been sent with a single request. At scale this would be a problem.

Let's assume you have 1,000 posts that are the max post length, which would be 128KB per post. That would be about 125MB for all 1,000 posts if they were each a single request. But we must multiply that times 151,552 since we are doing a request per character. This puts us at *18 Terabytes of data*. This extra bandwidth will incur an unnecessary additional cost to both ShlinkedIn and the users as well, especially to those on phones using mobile data.

I know the typical user doesn't submit a 4,000 character post, but even a 326 (which I feel is fairly representative of shlinkedin posts) character post uses 4.25 more data than is needed with a single request.

Even assuming smaller post lengths the current method is highly inefficient and correcting it will probably have some level of improvement for the server.

This also effects the CPU of the server. Each server/client interaction uses up CPU cycles. I can't do the math on this since I don't have any stats on the Phoenix server, but since a server/client message are sent for each character that represents 8,000 mesages for a 4,000 character post. All when it could have been 2 messages between the server/client. This will effect both memory and CPU usage, and I'm worried that at scale we will burn through servers quickly.

Sorry for the long post, and if you have any questions about my methods or my math, feel free to ping me.