CPS 373 Homework 1

Christopher Chapman (cchapman@fandm.edu)

October 28, 2020

1. Fetching HTTP Resources

Procedure
Each html request and response has a 26 byte header.

1. Request index.html (26 bytes)
2. Respond with index.html (5240 + 26 bytes)
3. Request header.html (26 bytes)
4. Request portrait.png (26 bytes)
5. Respond with header.html (1200 + 26 bytes)
6. Respond with portrait.png (16,000 + 26 bytes)
7. Request logo.png (26 bytes)
8. Respond with logo.png (8400 + 26 bytes)

Total number of bytes sent: 31,048 B
Total number of bits sent: 31,048 * 8 = 248,384 b
Rate: 800 bits/second
Time for a client to fetch all resources: number of bits sent / rate = (248,384b) / (800b/s) = **310.48 s**

2. Network Stack Shuffle: Encryption in the Transport Layer

Encryption is normally handled inside the application layer, by the application itself. I think it would be interesting to consider adding an encryption mechanism in the transport layer. In this model, applications can still encrypt the data they want to send, but it will be encrypted again by some mechanism within the transport layer. This can help offload encryption tasks to the OS with some encryption transport protocol. This allows any software with a network component to security transmit information. I'm sure this would not be easy to implement, and it may not be as secure as application layer encryption, but this would ensure that all data sent over networks will always be encrypted.

After writing this, I did some googling and looked a little closer in the book and it looks like this is already implemented using TLS/SSL protocol. Although it looks like this is still in the application layer, but works seamlessly with TCP. Perhaps the argument could be made to natively implement encryption into TCP. This would, by default, encrypt all data being sent over a network using TCP.