Jainit Bafna

2021114003

A. Differences in TCP Implementations

1. Traditional TCP starts with a 3-step setup process when connecting, but we skip this step.
2. TCP uses a “SYN” signal for synchronization, but we don’t use it.
3. In TCP, acknowledgments are sent after receiving a few pieces of data, but we send an acknowledgment for every piece we get.
4. We don’t have something called “flow control,” which helps manage data speed differences between devices.
5. Both our system and standard TCP allow data to go back and forth, but we need data to go in a specific order from the client to the server and vice versa.
6. Unlike normal TCP, we don’t use certain flags like “FYN” and “RST” for ending connections.

B. Incorporating Flow Control

To make sure data moves smoothly, we can do the following:

1. After connecting, the client tells the receiver how much data it can send.
2. The receiver checks how much it can receive and tells the client the lower value, which becomes the “window size.”
3. The client adjusts the size of the data it sends to match this “window size.”

For more flexible control, we can:

1. Instead of dividing data in advance, we send a fixed chunk size and keep the rest.
2. We measure how long it takes to send a piece and get a “I got it” message from the other side.
3. If it takes too long, we send smaller pieces next time.
4. If it’s pretty quick, we can send bigger pieces.

This way, we can manage how data moves to keep things running smoothly.

- The client adapts the size of transmitted data to align with the defined "window size."

For more flexible control, the following methods can be employed:

1. Instead of pre-determining data division, a fixed data chunk is sent, and the remainder is retained.
   - Rather than pre-dividing data, we transmit a fixed data chunk and keep the remaining data.
2. Measurement of the time it takes to send a data piece and receive an "I got it" message from the other end.
   - We gauge the time required to transmit a data piece and receive an acknowledgment in the form of an "I got it" message from the receiving end.
3. If the transmission takes too long, smaller data pieces are sent in subsequent transmissions.
   - If data transmission is time-consuming, smaller data pieces are dispatched in subsequent transmissions.
4. Conversely, if the process is quick, larger data pieces can be transmitted.
   - Conversely, if the transmission is rapid, larger data pieces can be sent.

This approach enables effective management of data transfer for smoother operation.