**Networking Assignment 5**

**Name: Eimantas Pusinskas**

**Student ID: 120312336**

**Stop-and-wait**

- the sender; sends a packet one at a time, once it receives an acknowledgement from the receiver for receiving a packet it sends the next packet and so on

- the receiver; receives a packet from the sender once the last bit from the packet arrives it sends an acknowledgement back to the sender

- problems that may arise include packet loss, ACK loss and premature timeout

- the throughput of stop-and-wait is one packet per round-trip

**Pipelining**

- Pipelining is similar to stop-and-wait but instead of sending one packet at a time you can send n packets consecutively and then receive their acknowledgements consecutively

- the throughput should be n, where n is the number of packets, but if n must be less than the receivers buffer size

**Go-Back-N**

- the sender; has a sequence of consecutive packets and a window of size N, where the window specifies the packets to be sent in the sequence. Once all packets in the window have been sent and acknowledgements have been received for each packet, the window slides right making the first packet sent previously no longer in the window and the next packet in the sequence as the last packet in the window

- the receiver; receives packets and sends and acknowledgement back for every correctly received packet which is determined by checking if the expected sequence number matches the sequence number of the packet received

**Conclusion**

In my opinion, pipelining is the best option. Stop-and-wait is fairly inefficient and pipelining improves on this by increasing the quantity of packets being able to be sent simultaneously. Go-Back-N in my opinion does an unnecessary amount of work by sending the same packets N times and potentially having to send multiple acknowledgements for the same packets. Thus I feel like pipelining is the best option, it is more efficient than stop-and-wait and does less work than Go-Back-N for the same outcome