Team Programming Assignment: Reliable Transport Protocol

Design Document

Authors: John Phillips, Matthew Lavengood, James Schreiner

Alternating Bit Protocol:

- Sending From A:
    - Make the packet containing the data, sequence, and checksum
    - Send the packet
    - Start the timer
    - Update A state
- Receive From B at A (wait phase):
    - Check if package is corrupt
    - Check if the sequence is the correct sequence
        - If corrupt or incorrect sequence:
            - Resend message and start A timer
    - If no message occurs within the timer limit, resend from A and restart timer
    - If the packet is not corrupt and the correct sequence, stop the timer.
    - Update A state
- Receive From A at B:
    - Check if corrupted
    - Check sequence number
        - If the sequence number is incorrect or is corrupted:
            - Send NACK to A with n+1 sequence number.
    - If the sequence number is correct and the packet is not corrupt:
        - Send ACK to A.
    - Update sequence number.

Go-Back-N:

- Sending from A:
    - Try to buffer the packet regardless of status. If buffer overflow occurs, exit the function.
    - If next sequence number is less than window size (base + N)
        - Get a packet from buffered packets
        - Set packet fields with the next sequence number, data, and checksum
        - Send the packet to layer3
        - If the base equals the next sequence number:
            - Start our timer for A
        - Increment next sequence number

- Timeout at A:
    - Start the timer
    - Send all packets currently in our window (Starting at base and going to next sequence number - 1)
- Receiving at A from B:
    - If packet is corrupt, we do not update our base
    - Otherwise:
        - Base is equal to the packet acknowledgement number + 1
        - If base equals the next sequence number, we stop the time.
        - Otherwise, start the timer.
- Receiving at B from A:
    - If packet is not corrupted and has a sequence number equal to the expected sequence number:
        - Extract the data message from the packet
        - Pass the message to layer 3
        - Create the acknowledgement packet with our expected sequence number, acknowledgement number, and checksum
        - Send the packet back to A
        - Increment the expected sequence number
    - If it is corrupt or has an invalid sequence number, we send the default packet with our last correct sequence number